TABLE OF CONTENTS

Section Page

TABLE OF CONTENTS (CONTINUED)

Section Page

TABLE OF CONTENTS (CONCLUDED)

Section Page

LIST OF ILLUSTRATIONS

Figure | Page

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

NASA Contractor Report 2923

Environmental Effects From SRB Exhaust Effluents - Technique Development and Preliminary Assessment

A. I. Goldford, S. I. Adelfang, J. S. Hickey, S. R. Smith, R. P. Welty, and G. L. White

CONTRACT NAS8-31306
NOVEMBER 1977

NASA

NASA Contractor Report 2923

Environmental Effects From SRB Exhaust Effluents - Technique Development and Preliminary Assessment

A. I. Goldford, S. I. Adelfang, J. S. Hickey,
S. R. Smith, R. P. Welty, and G. L. White
Science Applications, Inc.
Huntsville, Alabama

Prepared for
George C. Marshall Space Flight Center
under Contract NAS8-31806

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1977

FOREWORD

This final report is submitted to Aerospace Environment Division, Space Sciences Laboratory, Science and Engineering Directorate, NASA-Marshall Space Flight Center, Alabama, in partial fulfillment of requirements under Contract No. NAS8-31806.

Science Applications, Inc., is indebted to Dr. Briscoe Stephens, Atmospheric Diffusion/Environmental Assessment Task Team, Aerospace Environment Division, for the technical guidance and useful suggestions in planning the revision to the NASA/MSFC Multilayer Diffusion Models and in the development of the objective meteorological parameter selection concepts. The suggestions and the assistance of the following are gratefully acknowledged: the H. E. Cramer Company - diffusion modeling; the IIT Research Institute - scavenging and washout data; Dr. Ronald Dawbarn of ARO, Inc. - Al_2O_3 experiments; NASA Langley Research Center - measurements and laboratory chemistry; McDonnell Douglas Corporation - Delta Thor parameters; Chemical Systems Division of United Technologies Corporation - Titan III booster parameters; U.S. Air Force Weather Service and their range contractor, Pan American World Airways, Inc. - meteorological data collected at Kennedy Space Center. The work under this contract is under the direction of Arnold I. Goldford, Science Applications, Inc.

1. INTRODUCTION

This final report discusses efforts made on a study over the last nine months which had as its primary objective the development of techniques to determine the environmental effects from the Space Shuttle SRB exhaust effluents. This determination required the blending together of a team which had diverse skills and capabilities. The study required personnel with experience and knowledge in propulsion chemistry, meteorology, computer technology, and fluid dynamics. The study utilized four different computing systems; the NASA REEDA, the NASA UNIVAC 1108, the NASA IBM 360, and the SAI IBM 370. Knowledge of their operating systems and details of similarities and differences between the machines' data storage, instructions, and peripheral equipment operations was required during the study. Intimate knowledge of meteorological data reduction methods was a necessity. The start of the development of the new cloud rise model required expertise in fluid dynamics. To obtain the source terms for the cloud rise calculations required a state-of-the-art analysis of the SRB and its exhaust effluents.

This study has developed many new and needed tools for the determination of the environmental effects from the Space Shuttle SRB exhaust effluents. A preliminary climatological assessment has been performed which will be used to guide the future full scale climatological assessment. The exhaust effluent chemistry study has been performed and the exhaust species have been determined neglecting several possibly important effects. A reasonable exhaust particle size distribution has been constructed which can be used for the deposition model. The effects of scavenging and absorption have not been included in the preliminary climatological assessment. The basic conclusion that can be drawn regarding the entire study is that the team has now done their homework, understands the complete problem more fully, has developed the required algorithms, learned the required technology, and is now able to perform a meaningful climatological assessment with the operational REED Description which can yield the required answers about the environmental effects from the Space Shuttle SRB exhaust effluents. These algorithms have not been interfaced into the REED Description.

Section 3 on the exhaust chemistry and Section 6 on the numerical cloud rise model are efforts funded under NASA Contract NAS8-31851. The partial results have been included in this report so that a reader can get a clear picture of the overall effort. It should be noted that the basic studies have been conducted with a Titan type vehicle having all solid propellant motors and not the Space Shuttle type vehicle which has both solid and liquid propulsors. The technology for the problem has been learned but the models must be tuned for the Space Shuttle and its unique characteristics.

This study performed and used the results of a preliminary climatological diffusion assessment to define the problems involved in performing a full scale assessment; therefore, *these preliminary air quality results should be used with extreme caution in drawing conclusions* regarding the environmental effects of the Space Shuttle exhaust effluents.

2. CLIMATOLOGICAL ASSESSMENT

Environmental impact evaluation will be based on calculations of the ground level concentrations using the NASA/MSFC Rocket Exhaust Effluent Diffusio.. (REED) Description (1,2) input data for each selected meteorological regime. The use of the REED Description for environmental assessment requires a detailed knowledge of the surface mixing layer. The thermodynamic and kinematic properties of this layer can be measured with radiosondes, tetroonsonde, and other instrumental platforms. Large samples of these data are required for a climatological assessment of environmental impact. The only data set available which is sufficiently large to satisfy this requirement was obtained from radiosondes. These data were obtained daily (at 0000Z and 1200Z) at KSC for more than fifteen years by the U.S. Air Force Air Weather Service. In addition, four soundings per day were taken during a five year period (1962 through 1966).

The tapes containing the radiosonde data will be scanned and subsets of profiles will be established which correspond to the various meteorological regimes that were developed for air quality assessments by Stephens and Sloan (3). These data subsets will ultimately be used as input to the REED Description for calculation of air quality impact.

The data to be used will be the KSC soundings from the period 1962 - 1966[*]. The sample cumulative probability distribution of maximum ground-level concentrations attributed to each meteorological regime will be calculated; these probability distributions will be useful for estimation of the probability of exceeding a specified maximum concentration for a particular regime.

[*] The data tapes were obtained from the U.S. Air Force Range contractor, Pan American World Airways.

2.1 SELECTION OF AVERAGE YEAR

Monthly average surface data during the subject period (1962 through 1966) were used for determination of the year which was most representative of normal conditions at KSC. Because of the convenience of obtaining the required summaries from regular NOAA weather stations, climatological data from a similar coastal location (Daytona Beach) 50 miles from KSC were used to represent KSC.

The criterion for selection of a particular year was that it have the smallest value of the parameter D given by

$$D = 1/12\left[\sum_{i=1}^{12} |T'_{Mi}| + \sum_{i=1}^{12} |T'_{mi}| + \sum_{i=1}^{12} |v'_{mi}|\right]$$

where i=1 corresponds to January, i=2, February, etc. and $|T'_{Mi}|$, $|T'_{mi}|$ and $|v'_{mi}|$ are the absolute deviation of the monthly mean daily maximum and minimum temperature and monthly mean wind speed from their respective normal monthly means; the quantity D represents the average monthly total absolute deviation for the three parameters.

The calculated values of D are given in the table below.

Year	62	63	64	65	66
D	4.85	5.30	4.83	3.41	3.65

Thus, the year 1965 was selected as the year most representative of normal conditions at KSC.

In connection with our selection of a climatological data set, the following background data for KSC were acquired from the National Weather Records Center:

- Monthly and annual inversion statistics for the period 1965 through 1969 based on KSC Rawinsonde data
- Monthly and annual STAR summary of atmospheric stability for the period 1965 through 1969 based on Cocoa Beach surface data

- Monthly and annual mixing height statistics for the year 1965 based on KSC Rawinsonde data

With regard to the 1965 KSC Rawinsonde data tape, we intend to calculate the following statistics as a function of time of day:

- Distribution of atmospheric stability calculated between 1.2 to 1.5 km (4,000 to 5,000 ft) by taking the gradient of virtual potential temperature
- Distribution of the height of ground based inversions
- Distribution of wind speed at 1.2 km
- Distribution of wind direction at 1.2 km

These statistics are correlated with the diffusion potential of the ambient air at typical SRB cloud stabilization altitudes. The distribution of the height of ground based inversions is useful in the study of how often SRB clouds are expected to penetrate such inversions and thus become effectively isolated from the ground; ground based inversions are also responsible for the largest concentrations observed at ground level whenever there is a release from a non-buoyant ("cold") source. The distribution of wind direction at the typical height of SRB cloud stabilization chosen (1.2 km) is correlated with the expected track of the SRB cloud at the calculated stabilization height.

2.2 METEOROLOGICAL REGIMES

In support of air quality assessments for aerospace vehicle exhaust effluents at Kennedy Space Center, meteorological regimes were defined which correspond to synoptic patterns (3). These regimes are designed to narrow the air quality statistics into categories that reflect temporal development of atmospheric conditions at launch.

In the past the meteorological inputs to the NASA/MSFC REED Description have been based mostly on climatological statistics until about 12 hours prior to launch, at which time a deterministic forecast was made. An obvious drawback to this approach is that the statistical air quality assessment during the pending launch period, two or four days prior to launch, does not reflect atmospheric dynamics identifiable from current synoptic conditions.

Thus, the purpose of defining meteorological regimes in terms of synoptic conditions is to provide a realistic means of classifying subsets of the overall climatological data set for statistical air quality assessments. Since these subsets are more representative of developing atmospheric conditions during the pending launch period, the use of these subsets assures a smoother interface of the statistical air quality assessment with the deterministic assessment. Employing this classification system, the statistical assessment affords error bounds for the deterministic predictions.

It is necessary to consider the types of atmospheric data sources and the applications for which the results of the diffusion predictions will be utilized in order to define appropriate meteorological regimes. The amount of detail required in the atmospheric kinematics is dictated by the planned application of the diffusion prediction. Two extremes in applications are air quality and deployment predictions. If the diffusion predictions are to be utilized in support of air quality predictions to insure public safety, the detail in the atmospheric input parameter can be relaxed in favor of slightly conservative values which incorporates a safety factor. Since the desire is to identify any potential for an air quality problem, the exact location and concentrations are of secondary importance as long as the error bounds for these estimates have been determined and are reasonably conservative. For this application, routine radiosonde data are satisfactory since small spatial and temporal changes in the atmospheric kinematics can be neglected without a serious impact on the creditability of the results.

On the other hand, if the application for the diffusion prediction is to support the deployment of the cost-effective rocket effluent monitoring network, the resolution requirements of the atmospheric input parameters for the REED Description are very stringent. This increase of rigor is introduced by the need for exactness in the predicted transit path. In this case, local spatial and temporal changes in the atmospheric kinematics must be considered. This means that terrain effects and the land-sea interface effects must be known. Since the radiosonde provides predominately vertical information, other sources of data must be used to obtain horizontal-temporal information. In general, wind tower data are not adequate to totally support this requirement since the available information is limited to the surface boundary layer. Currently, the best source of local spatial - temporal information is a tetroonsonde (a constant level balloon with radiosonde) flown nominally at 600 meters (4). Other potential means to obtain or improve the local spatial - temporal information would be from simultaneous multiple radiosonde releases or a remote sensing system. Hence exactness in predictions of the exhaust cloud transit path is limited by the state-of-the-art of the available small scale atmospheric measurement system. Extensive meteorological support of the NASA rocket exhaust effluent prediction and monitoring program have been documented for a series of seven Titan launches (5-11); the hydrogen chloride measurements for the same series are described by Gregory, et al. (12)

A common requirement for a diffusion prediction is the statistical air quality assessment for planning activities prior to a launch. The objective is to use these statistical assessments for mission planning activities to optimize launch windows.

Meteorological regimes needed for air quality assessment prior to launch were defined. The regimes were not intended for detailed launch effluent monitoring support.

Before defining the meteorological regimes, consideration of the selection and sequential nature of the approach will be described. Typically, there are about nine different patterns that could be associated with the weather conditions at Kennedy Space Center. Within each pattern, there are a wide variation in the small scale kinematic and thermodynamic structure depending on the type and intensity of the mesoscale activity present.

It is appropriate to use existing knowledge of seasonal variation at KSC in the selection of seasonal time regimes for statistical analysis. It is apparent that the length of the seasons at KSC are non-uniform with relatively long summers and winters (mid-May through mid-October and December through March, respectively) which are separated by short (approximately 6 weeks) transition periods. It is known that the summer and winter diffusion meteorology will contribute to the largest variation between calculated seasonal environmental impacts; since the realistic seasonal breakdown of data sets increases the size of the winter and summer sample it follows that the comparison of winter and summer will have better statistical reliability.

The approach is to start with the statistical air quality assessment that is normally used in the mission planning activities; initially the seasonal-temporal regimes are defined; that is, the season of the year--winter, spring, summer, or fall--and the time of day--night, morning, afternoon, or evening. Further narrowing of the regime categories will be achieved by sub-division into the following synoptic patterns:

- The Bermuda anticyclone and associated easterly winds.

- Easterly waves and associated strong vertical mixing.
- Westerly waves and associated frontal activity.
- Continental anticyclones and associated northerly winds.

The next step in the process is the qualification of the intensity of the synoptic regime according to nominal "weak" and "strong" categories. Objective criteria will be established for such a qualification.

In summary the regimes established will consist of the following categories:

- Season
- Synoptic regime
- Intensity of synoptic regime
- Time of day

Other regime categories such as thermodynamic or kinematic parameters may be better suited for climatological air quality assessments.

2.2.1 Air Quality Impact and Associated Meteorological Patterns

Air quality impact can be classified according to concurrent synoptic meteorology patterns and air mass types. The relative frequency of occurence of these patterns during 1965 at KSC has been calculated. NOAA synoptic charts drawn twice daily (1 a.m. and 1 p.m. EST) were used for the analysis. The following synoptic and air mass classification was used:

Synoptic Type	Synoptic Class	Air Mass
A	Maritime Anticyclone	Maritime Tropical (MT)
B	Easterly Wave	Maritime Tropical (MT)
C	Westerly Wave	[1]Maritime Tropical Continental Polar (P) Transition (MT-CP)
D	Continental Anticyclone	Continental Polar (CP)

[1] Specification of the air mass type for Type C is dependent on the type and strength of the front (cold, warm, stationary) and its location relative to KSC.

This classification is essentially the same as that given in Reference 1 w th slight modification of synoptic Type A to represent the general category of maritime anticyclones which is composed of two seasonal sub-types. In summer the maritime anticyclone is synonymous with the Bermuda anticyclone which persistently dominates the weather in the Eastern United States. The only break in this persistent pattern occurs in late summer when inverted low pressure troughs embedded in the tropical easterlies move to the vicinity of KSC (Type E Easterly Wave). These troughs are in rare instances associated with a hurricane. In winter, anticyclones containing cold dry air move southeastward toward KSC; as these circulations pass over the relatively warm water east of the Florida peninsula, they are rapidly modified. Thus, in winter, there is a typical alternating pattern of Type A and Type D anticyclones. The transition between the two types is characterized by Type C (Westerly Wave) conditions which include clouds and precipitation associated with fronts and eastward propagating waves in the westerlies (Type C, Westerly Wave).

The monthly and annual percent occurrence of the various synoptic regimes and air mass types during 1965 is given in Table 2-1. It is clearly indicated that the predominant synoptic regime is the maritime anticyclone (Type A) with an annual occurrence of 57.6 percent; on a monthly basis Type A predominated during the period March through November. During the winter months (December through February) continental anticyclones are often strong enough to penetrate far enough southward to become the predominant synoptic Type D at KSC. The occurrence of air mass types is correlated with the occurrence of the synoptic types.

It is obvious from the analysis that the summer season is the most critical in the assessment of environmental

Table 2-1. Percent Occurrence of Synoptic and Air Mass Types at Kennedy Space Center During 1965

	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.	Annual
Synoptic Type													
A	27.4	35.7	43.5	66.7	88.7	90.0	90.3	86.8	60.3	37.1	43.3	19.7	57.6
B	0	0	0	0	0	0	6.5	6.6	20.7	8.1	0	0	3.4
C	14.5	19.6	30.6	6.7	1.6	6.7	3.2	6.6	19.0	29.0	21.7	26.2	15.4
D	58.1	44.6	25.9	26.7	9.7	3.3	0	0	0	25.8	35.0	54.1	23.6
Air Mass Type													
CP	58.1	48.2	29.0	20.0	4.8	0	0	0	0	22.6	20.0	50.8	21.1
MT	33.8	41.1	51.6	70.0	88.7	95.0	100.0	100.0	100.0	53.2	55.0	26.2	67.9
CP-MT[1]	8.1	10.7	19.4	10.0	6.5	5.0	0	0	0	24.2	25.0	23.0	11.0

Synoptic Types

A = Maritime Anticyclone
B = Easterly Wave
C = Westerly Wave
D = Continental Anticyclone

Air Mass Types

CP = Continental Polar
MT = Maritime Tropical
MT-CP = Transitional Air Mass

impact in populated areas west of KSC. During this period Easterly flow associated with the maritime (Bermuda) anticyclone will occur during a large percentage of the time. During the daytime, the synoptic flow is enhanced in the surface layer by the local sea breeze circulation. As the air associated with the sea breeze circulation moves onshore, a ground based mixed layer develops. The thickness of the mixed layer is a function of the intensity of turbulence generated by mechanical interaction of the air with the land surface roughness elements and land-to-air heat transfer. It is hypothesized that concentrations of SRB effluents may occur at ground level locations in areas west of KSC when portions of the stabilized SRB cloud are within the sea breeze mixed layer. This hypothesis will be tested in a planned study based on the available sample of Rawinsondes obtained during the period 1100 to 1500 EST during the summer months (June through September) of 1965. The sub-sample of soundings which exhibit a well-developed sea breeze and a mixed layer extending above the stabilized SRB cloud will be used as input data to the UNIVAC 1108 REED Description. The calculated maximum concentrations and dosages will be compared with those calculated at times of the year during different meteorological regimes and times of the day. If the hypothesis is verified for the 1965 data, a more detailed analysis will be initiated based on the additional summer soundings that can be drawn from the existing data tapes for the year 1962 through 1964 and 1966. The results of this study will comprise the maximum estimated impact, assuming that there are no launch constraints based on air quality impact considerations.

During summer nights, the land breeze will tend to be minimized, since it is opposed by the large scale synoptic flow; it is during this period when the flow is poorly organized that the forecasting of SRB cloud trajectory will be the most difficult. However, calculated downwind concentrations during the night are not expected to be as large as those during the day because of the decreased rate of vertical diffusion associated with the tendency of the atmosphere near the ground to become neutral or stably stratified during this period.

The idea that a representative sub-sample of meteorological data can be drawn from a larger sample was tested by comparing percent occurrence of synoptic and air mass types during 1965 for 102 cases (based on two NOAA synoptic charts per day at 1 a.m. and 1 p.m. EST for one day per week for 51 weeks) to that obtained for 726 cases based on twice daily data for 363 days. The results of this comparison are given in Table 2-2. It is indicated that the statistics of the sub-sample in most categories correspond closely to those of the parent sample. The only significant deviation is for the occurrence of synoptic Type C which is underestimated in the sub-sample. This can be attributed to the fact that Type C is a transient phenomena that is not accurately seen by weekly sampling.

Table 2-2. Percent Occurrence of Synoptic and Air Mass Types for the Parent Sample (726 Cases) and the Sub-sample (102 Cases) for 1965 at KSC

Synoptic Type	Air Mass	Parent Sample (726 Cases)	Sub-Sample (102 Cases)
A	MT	56.2	59.8
	CP	0.0	0.0
	MT-CP	1.4	1.0
	TOTAL	57.6	60.8
B	MT	3.4	3.9
	CP	0.0	0.0
	MT-CP	0.0	0.0
	TOTAL	3.4	3.9
C	MT	8.3	4.9
	CP	1.9	2.0
	MT-CP	5.2	1.0
	TOTAL	15.4	7.8
D	MT	0.0	0.0
	CP	19.1	24.5
	MT-CP	4.1	2.9
	TOTAL	23.6	27.5
TOTAL	MT	67.9	68.6
	CP	21.1	26.5
	MT-CP	11.0	4.9

2.3 PRELIMINARY RESULTS FROM 1969 METEOROLOGICAL DATA SPACE SHUTTLE CLIMATOLOGICAL DIFFUSION ASSESSMENT

A preliminary climatological assessment study was begun using 1969 meteorological data and effluent parameters given in the Agency Environmental Impact Statement for the Space Shuttle (13). A sample of 101 soundings (one day per week and 2 soundings per day) were generated from a 1969 met data tape using the met screening program. With the aid of the AEC and TVA stability criteria output by the program for each sounding, the height of the surface transport layer was chosen and input cards for the pre-processor were assembled. The 101 cases were then run through the multi-layer/pre-processor system and the results tabulated. Table 2-3 shows the two worst cases of the 101 processed. The November 16th case is further illustrated in Figure 2-1. Note that for January 8, the maximum dosage approaches the critical NAS level (2400 PPM-sec) as does the maximum peak concentration for November 16 (critical NAS level = 8 PPM)(14).

Table 2-3. Summary of Worst Cases from 1969 Sample of 101 Cases

Vehicle	Date	Time	Model	Pollutant	Adjusted Cloud Rise Height
Space Shuttle	01/08/69	12Z	4	HCl	979.18
Space Shuttle	11/16/69	12Z	4	HCl	1135.57

Range	Azimuth Bearing	Max Peak Conc.	Max Dosage	Max Peak 10 Min Time Mean Conc.
261.03	80.28	1.522	2176.083	1.450
1062.87	194.78	5.034	719.015	1.198

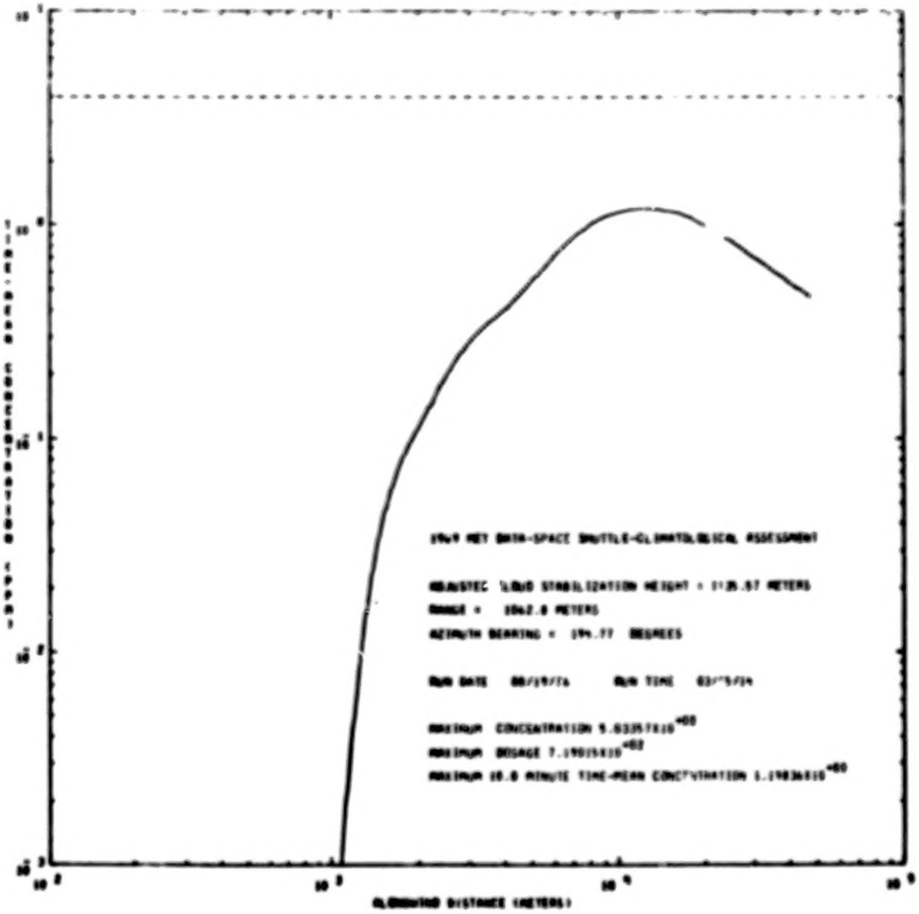


Figure 2-1. Maximum Predicted (Model 4) Ten-Minute Time Mean Ground-Level Centerline HCl Concentration (PPM) for a Normal Space Shuttle Launch (Rawinsonde Input Data for 11/16/69, 12Z).

Examination of data obtained from towers at various locations at KSC has indicated that surface* temperatures can be highly variable. This variability leads to a degree of uncertainty in the diffusion calculations, which are usually based on data obtained at one location. To illustrate this uncertainty the diffusion calculation for the worst case maximum centerline HCl concentration (11/16/69, 12 Z) was repeated using a revised surface temperature of 0^{o}C which was 7^{o}C colder than the original temperature. This temperature difference is within the expected range of variability of surface temperatures at KSC. The results are illustrated in Figure 2-1 (original calculation, surface temperature = 7.0^{o}C.) and Figure 2-2 (revised surface temperature = 0.0^{o}C). It is shown that the revised maximum concentration increased to 6.92 PPM from the original value of 5.03 PPM; the maximum dosag ncreased from 719 to 831 PPM-sec for the revised data. It is concluded that surface temperature uncertainties in the input meteorological data lead to uncertainties in the calculated air quality impact. Other workers have indicated an uncertainty of as large as a factor of two in the diffusion model results, largely attributable to meteorological uncertainties. However, field measurements taken after TITAN launches (1) suggest a significantly smaller uncertainty (10 to 25 percent).

In view of the uncertainties in the calculations and the limited sample of KSC meteorological data uses, the results for peak concentration described below are considered very preliminary.

By comparing the peak concentration data to NAS standards the following categorization scheme was devised by Dr. Stephens for mapping of the results. Future results based on a large data sample will use the color categories given below:

* Actual height about 2m above surface.

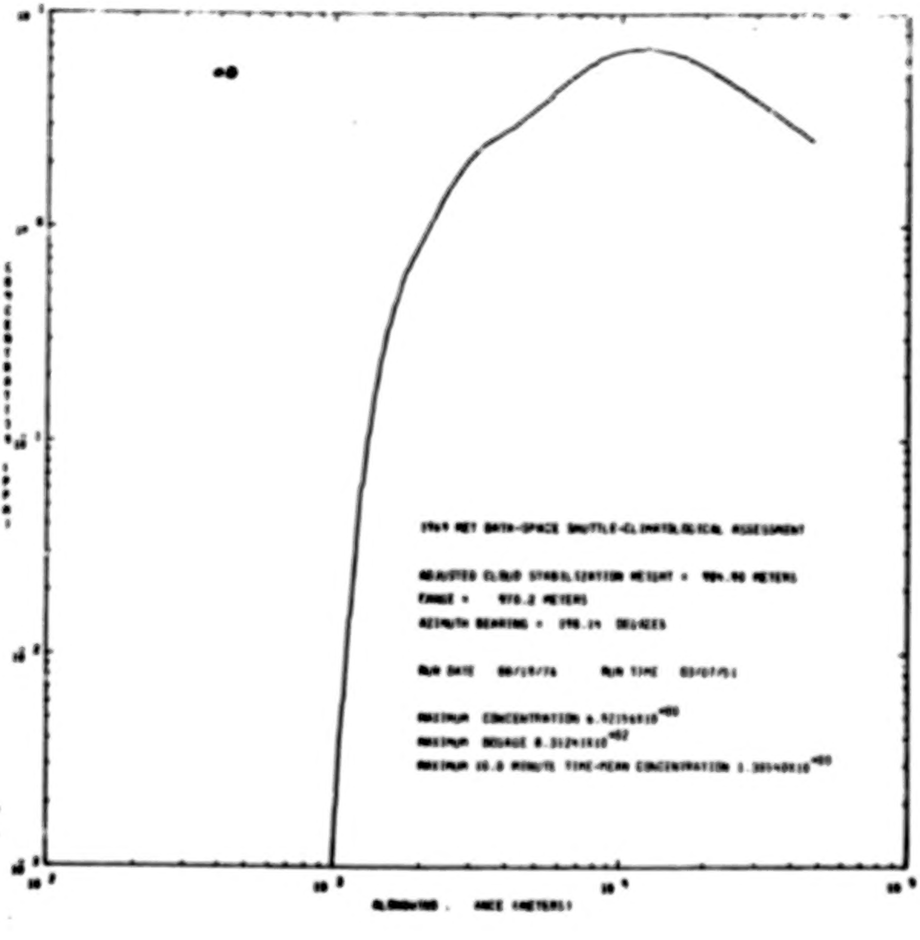


Figure 2-2. Maximum Predicted (Mode. 4) Ten-Minute Time Mean Ground-Level Centerline HCl Concentration (PPM) for a Normal Space Shuttle Launch (Rawinsonde Input Data for 11/16/69, 12Z Modified for Model Sensitivity Test; Surface (16 Ft) Temperature Reduced by 7°C).

COLOR	MAXIMUM PEAK CONCENTRATION (PPM)	% of CASES*
Green	$\leq$ 4.00	96%
Yellow	4.01 to 5.00	3%
Orange	5.01 to 8.00	1%
Red	> 8	0

* Based on the 101 cases during 1969 at KSC.

2.3.1 Air Quality Guidelines

The climatological air quality assessment of the impact of the Shuttle SRB exhaust cloud requires the comparison of ground-level concentration and dosage predictions to air quality guidelines given by the National Academy of Sciences (NAS), with the exception of industrial standards applicable to KSC which assume chronic exposure. There are no national standards for the short-term exposures associated with aerospace exhaust effluents (Ref 14). A graphical illustration of how a statistical summary of dosage predictions can be compared with an NAS guideline for aerospace applications is given in Figure 2-3. The particular NAS guideline used in the illustration is the short-term public limit for a 10-minute average exposure (STPL 10) which is 4 parts per million (ppm) for HCl with an 8 ppm ceiling. This is equivalent to a dosage of 2400 ppm-sec. The cumulative distribution of maximum 10-minute dosages (expressed in percent of 2400 ppm-sec) predicted by the NASA/MSFC REED Description for 101 cases during 1969 is plotted in Figure 2- 3. It is shown that 98 percent of the predicted dosages were less than 34 percent of the NAS standard. The largest predicted dosage was 2176 ppm-sec (January 8, 1969), which was 91 percent of the NAS standard. These results are preliminary. Additional calculations, based on an updated diffusion model, the objective methods for specification of the standard deviation of wind azimuth angle (SIGAR) and transport layer height, and the large sample of data available for 1965 (<1400 cases) will be made as the study continues.

Initial indications suggest that the Space Shuttle does not have an air quality problem under normal atmospheric conditions. However, marginal air quality conditions could exist within KSC which could result in a requirement for crowd control.

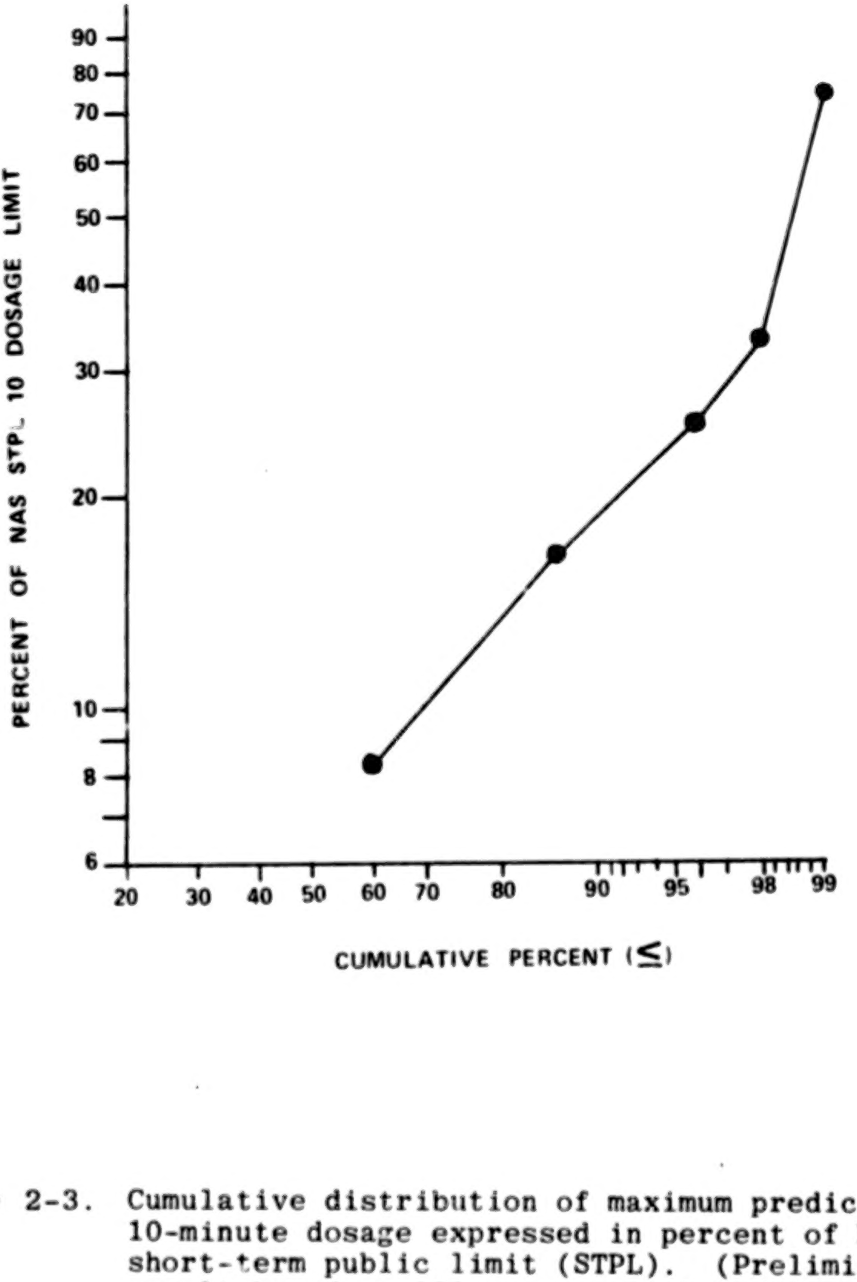


Figure 2-3. Cumulative distribution of maximum predicted 10-minute dosage expressed in percent of NAS short-term public limit (STPL). (Preliminary result based on 101 cases calculated from KSC Rawinsonde data at 00Z and 12Z)

2.3.2 Transport Direction of the Stabilized SRB Cloud

The statistics of expected transport direction of the stabilized SRB exhaust cloud give an indication of the direction where significant impact is most probable (assuming no meteorological launch constraints with regard to expected air quality impact). A preliminary evaluation of the distribution of SRB exhaust cloud transport, based on 101 cases in 1969, is illustrated in Figure 2-4. The transport direction was estimated from Rawinsonde data by taking the wind direction at the altitude nearest the cloud stabilization height. In more than 70 percent of the cases this altitude was within 100 meters of the cloud stabilization altitude. For the other cases, examination of the wind direction profile did not justify interpolation to obtain a better estimate of wind direction. Transport direction is taken as 180 degrees plus the wind direction. Thus an east wind (90°) results in a westward transport direction (270°). It is shown that the transport directions with the largest calculated frequency of occurrence (12 precent) were east-southeast and northwest. Further comments on transport direction statistics are reserved for forthcoming calculations based on larger data samples. The distribution of transport direction at KSC will be derived as a function of time of day (0100, 0700, 1300, 1900 EST) for the 1965 transport directions for each time of day that can be obtained using the 1965 data.

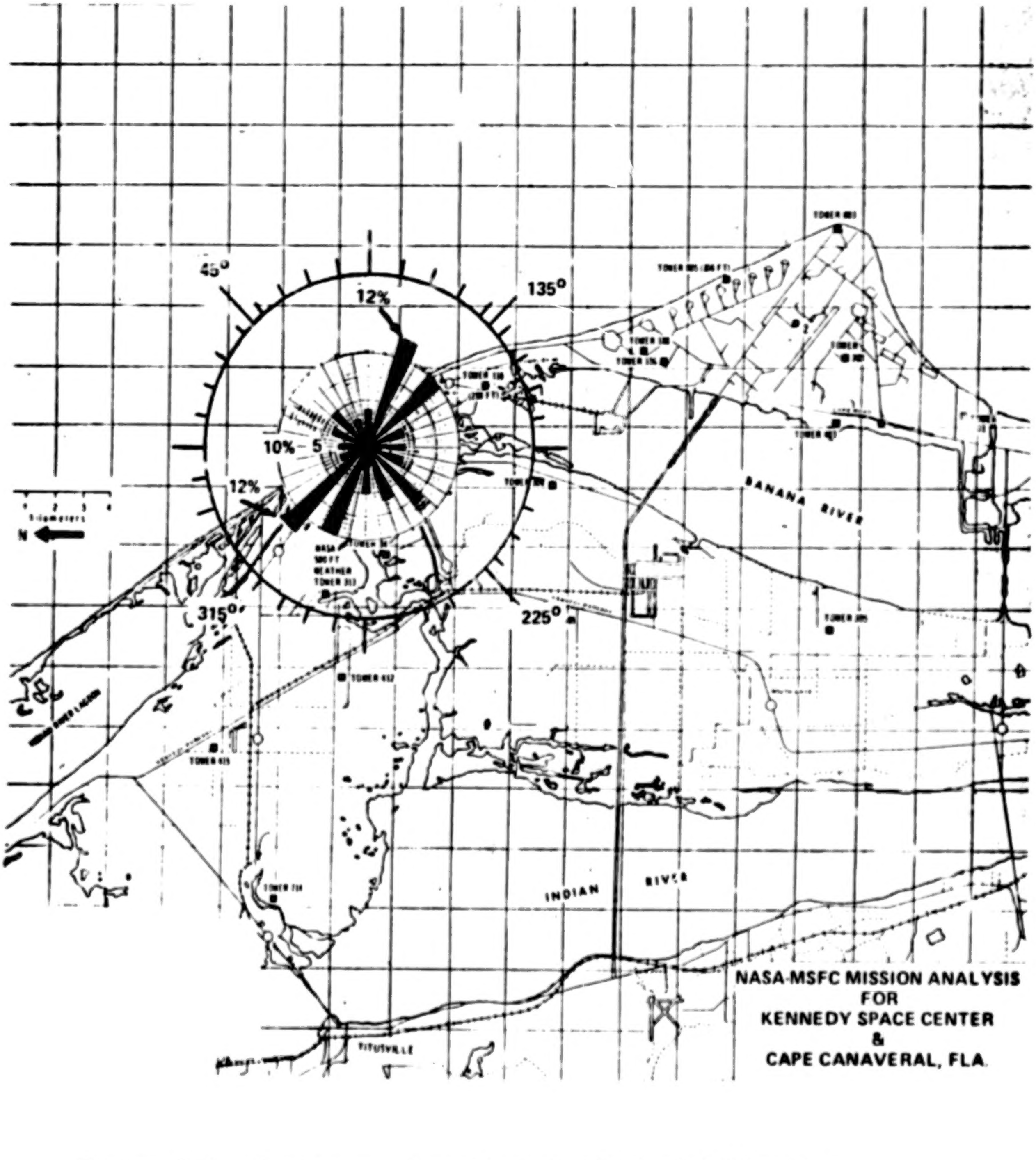


Figure 2-4. Transport direction at cloud stabilization height expressed in percent occurrence. Based on weekly Rawinsonde data obtained twice daily (00Z, 12Z) during 1969 (101) cases).

2.4 NEW OBJECTIVE CONCEPTS

A portion of this study effort was spent in exploring new concepts for the objective analyses of the meteorological data. These objective analyses, if proven by theory and test, would allow the automatic selection of REED Description parameters. That would lead not only to a large savings in manpower but to a better, more uniform treatment of the data.

2.4.1 Transport Layer Height Determination

An attempt to develop objective criteria for selection of the transport layer height used in the NASA/MSFC Multilayer Diffusion Model has been made. Acceptable criteria will permit the development of a computer code for the automation of transport layer height selection. Although an acceptable set of objective criteria have not yet been found, preliminary criteria have been established and are being tested.

Two sets of criteria listed in Tables 2-4 and 2-5 were studied. The relative frequency of occurrence of the various transport layer categories are also given in the tables. In the first set of criteria, outlined schematically in Figure 2-5, strong emphasis is placed on the existence of stable layers below the cloud. This results in a large number of transport layer heights below cloud stabilization height, which in effect reduces calculated ground level concentrations by reducing the amount of cloud mass which can be diffused downward. If these stable layers are proven to have a smaller frequency of occurrence because of inaccuracies of the Rawinsonde data or are not related to actual transport layer heights, the calculated air quality impact will not be conservative. In the present stage of development of our capability to predict air quality impact, it is not desirable to use techniques that may later be proven unconservative.

The criteria listed in Table 2-5 will give conservative results because emphasis is given to the occurrence of wind

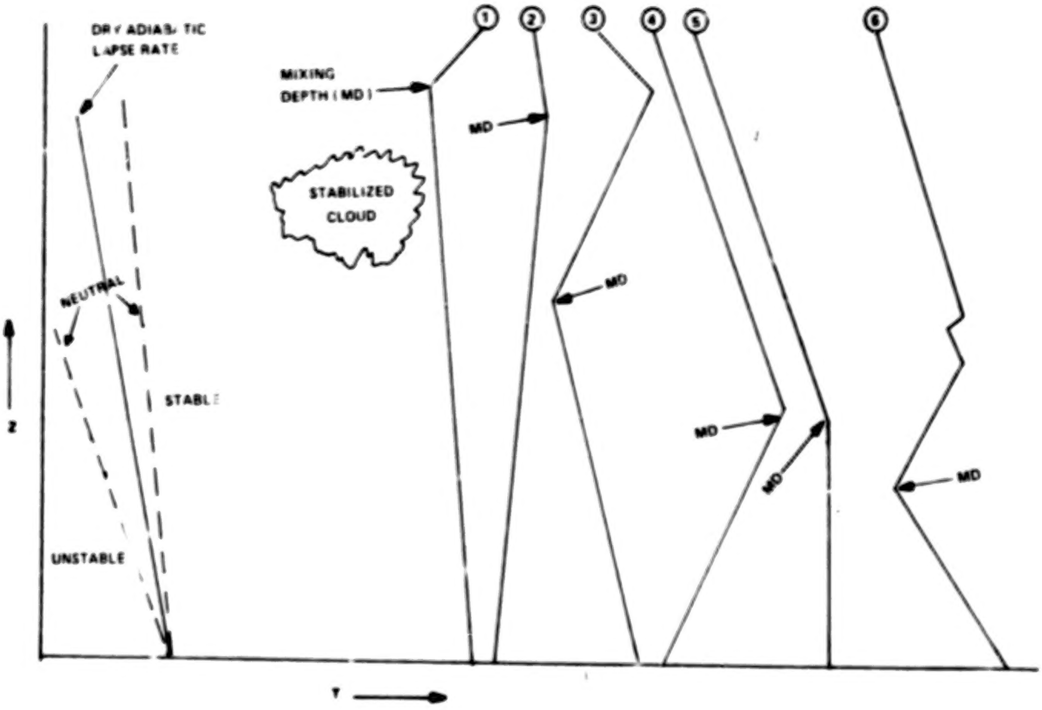


Figure 2-5. Typica] Temperature Soundings for Various Mixing Depth Categories of Transport Layer Height Given by Circled Number.

Table 2-4. Transport layer height percent occurrence in six categories derived from weekly Rawinsondes at KSC during 1969.

	CATEGORY	PERCENT OCCURRENCES		
		1200Z (0700 EST) 50 cases	0000Z (1900 EST) 51 cases	Combined 101 cases
1.	Base of stable layer above cloud; no stable layers or inversions at or below the cloud altitude	10.0	7.8	8.9
2.	Top of ground based stable or inversion layer in which the cloud is immersed	0.0	3.9	2.0
3.	Base of stable or inversion layer in which the cloud is immersed; no stable or inversion layers below	10.0	11.8	10.9
4.	Top of ground based inversion beneath the cloud	48.0	21.6	34.7
5.	Top of ground based stable layer$_{(A)}$ beneath the cloud	24.0	21.6	22.8
6.	Base of lowest stable layer beneath cloud	8.0	33.3	20.8

(A) Category 5 is synonomous with category 4 when the stable layer extending upward from the ground consists solely of a temperature inversion.

Table 2-5. Transport layer height percent occurrence in five categories derived from weekly Rawinsondes at KSC during 1969.

	CATEGORY	PERCENT OCCURRENCE		
		00Z 1900 EST 51 cases	12Z 0700 EST 50 cases	Combined
1.	Base of stable layer above exhaust cloud	64.7	64.0	64.4
2.	Top of wind shear layer	19.6	8.0	13.9
3.	Base of wind shear layer	0	2.0	1.0
4.	Top of surface based stable layer with potential temperature gradient >.0098°C/meter extending to altitudes >250 meters	13.7	8.0	10.9
5.	Top of stable layer in which cloud is immersed	2.0	4.0	9.9

shears and stable layers above cloud stabilization height.

A code has been written which executes the logic developed by the H. E. Cramer Co. for selection of the height of the surface transport layer; selection is based on criteria for the vertical gradient of virtual temperature $\Delta T_v/\Delta z$ summarized below:

- A ground based inversion is defined if

  $\Delta z \geq 100\text{m}$ and $\dfrac{\Delta T_v}{\Delta z} \geq -.0005$ °C/m

- The base of a stable layer at z_1 if

  $\Delta z = z_2 - z_1 \geq 100\text{m}$ and

  $\dfrac{\Delta T_v}{\Delta z} \leq -.005$ °C/m

  where $z_2 > z_1$

If a ground based inversion exists, the height of surface transport layer is specified as the top of the ground based inversion; otherwise it is the height of the base of the first stable layer above the ground. If the base of the first stable layer is above 3000m the depth is set equal to 3000m.

The code will be used for specification of the surface transport layer for the 1965 Radiosonde data (>1400 cases). The calculation of transport layer height, H_m, will be made concurrent with the calculation of the stabilization height, H_s, of the SRB cloud. A criteria will be established to identify cases when calculated downwind concentrations and dosages are essentially zero. These cases are associated with a very low transport layer height relative to cloud stabilization heights. The criteria would be of the form

$$H_s - H_m \geq X$$

where X would be selected on the basis of a sub-set of diffussion calculations for various values of $H_s - H_m$. The results of these calculations can be illustrated hypothetically as shown in Figure 2-6; based on this hypothetical example X would be chosen to be 400 meters.

The criteria would be used to eliminate trivial cases from the large parent sample.

2.4.2 Bivariate Normal Wind Distribution

In addition to the other analyses of the 1965 data, a study of the theory of the bivariate normal distribution and it's use in summarizing the wind statistics at KSC was conducted. The theoretical equations and derivation supplied by O. E. Smith of NASA/MSFC were checked out, and the essential equations have been coded. Given the bivariate normal statistics, the program outputs the following:

- The distribution of wind direction.
- The distribution of wind speed given a specific direction (15).

Since the monthly bivariate normal statistics have already been calculated for KSC, the programs developed would be used as part of our operational forecasting scheme.

2.4.3 Development of Objective Methods for Estimation of Meteorological Input Variables for the Multilayer Diffusion Model

The development of objective methods for the estimation of meteorological input variables for the Multilayer Diffusion Model requires the following:

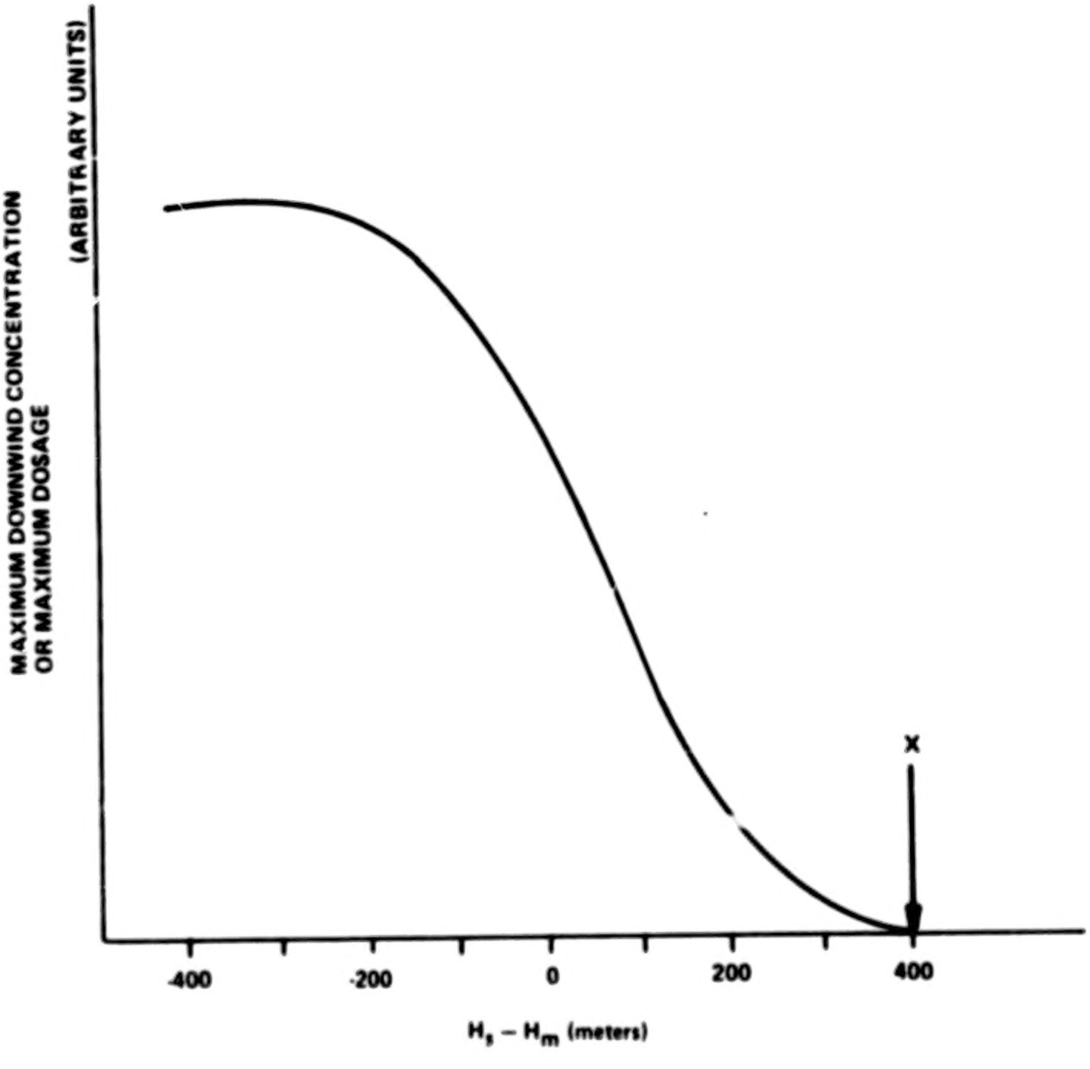


Figure 2-6. ⁷ypothetical schematic represe⁻tati⁻a of relation between air quality impact ₋f SRB cloud and the difference betw₃en cloud stabilization height and transport layer height (H_s - H_m).

- Establishment of a theoretical basis for the method,
- Development of computer codes for calculation of the standard deviation of wind azimuth angle over a ten-minute period (SIGAR) and for specification of height of the transport layer (H_m),
- Testing the computer codes for a climatological data sample,
- Modification of the preprocessor to include the new codes.

The theoretical basis (16) for the method selected for estimating SIGAR is based on solution of equation

$$\text{SIGAR} = \sigma_v/U = \frac{kf(B)}{\ln z/z_o - \psi(RI)} \qquad (1)$$

where σ_v = standard deviation of the lateral component of turbulence (m/sec)

U = mean wind speed (m/sec)

z_o = roughness length (m)

k = Von Karman's constant = 0.4 (dimensionless)

The function of Richardson number, $\Psi(RI)$, for unstable conditions is

$$\Psi(RI) = 2 \ln \left[(1+x)/2\right] + \ln \left[(1+x^2)/2\right] - 2 \tan^{-1} x + \pi/2 \qquad (2)$$

where $x = (1-16RI)^{\frac{1}{4}}$ (2a)

For stable conditions (17)

$$\Psi(RI) = \frac{7\ RI}{1-7RI} \qquad (3)$$

The right side of Equation 1 was derived (16) by subsitution of expressions for σ_v and U given below for the ratio σ_v/U.

$$\sigma_v = \mu^* f(B) \qquad (4)$$

$$U = \mu^*/k \left[\ln z/z_o - \Psi(RI)\right] \qquad (5)$$

where μ^* is the friction velocity and the function f(B) is accurately approximated by fitting line segments to experimental measurements of the ratio σ_v/μ^* according to,

f(B)	B	
2.7	$B < -.008$	
2.7 + 112(.008+B)	$-.008 \leq B < -.00175$	
3.4 - 725.5(.00175+B)	$-.00175 \leq B < .0008$	6
1.55 + 38.04(B-.0008)	$.0008 \leq B < .029$	
2.35 + 5.43(B-.029)	$.029 \leq B$	

The Richardson number, RI, is defined by

$$RI = \frac{\frac{g}{t}\frac{\partial\theta}{\partial z}}{\left(\frac{\partial v}{\partial z}\right)^2} \qquad (7)$$

where g = acceleration of gravity (m/sec^2)

t = absolute temperature (^{o}K)

$\frac{\partial\theta}{\partial z}$ = vertical gradient of potential temperature (^{o}K/m)

$\frac{\partial v}{\partial z}$ = vertical gradient of wind speed (1/sec)

The quantity $\frac{\partial \theta}{\partial z}$ can be expressed as a function of pressure and vertical gradient of temperature according to

$$\frac{\partial \theta}{\partial z} \approx \frac{\Delta \theta}{\Delta z} = \left(\frac{1000}{P}\right)^{.288} \left(\frac{\Delta t}{\Delta z} + .0098\right) \tag{8}$$

where P is the pressure in millibars.

Since available wind measurements are not sufficiently accurate for estimation of the denominator in Equation (7), it is necessary to estimate RI from measurements of the non-dimensional stability ratio, B.

$$B = \frac{g\,\bar{z}^2}{tU^2}\ \frac{\Delta\theta}{\Delta z} \tag{9}$$

where, $\bar{z}$ = the geometric mean height (m) between the top and bottom of the layer considered (17)

U = mean wind speed in this layer. (m/sec)

The relation (17) between B and R_i is

$$RI = B\left[\frac{\ln z/z_o - \Psi(RI)}{\phi(RI)}\right]^2 \tag{10}$$

where, $\phi(RI) = (1-16RI)^{-\frac{1}{4}}$ for unstable conditions (11)

$\phi(RI) = \frac{1}{1 - 7RI}$ for stable conditions (12)

For stable conditions it can be shown that Equation (10) is a quadratic function of the parameter y.

$$y^2 + y/7kB^{\frac{1}{2}} - (k+1)/7k = 0 \qquad (13)$$

$$\text{where, } y = (RI)^{\frac{1}{2}} \qquad (13a)$$

$$k = \ln(z/z_o)-1 \qquad (13b)$$

For unequal real roots, the root given by the following equation will result in physically realistic calculated values of RI over the expected range of measured values of B.

$$y = -\frac{1}{14k\sqrt{B}} + \frac{1}{2}\sqrt{\frac{1}{49k^2B} + \frac{4(k+1)}{7k}} \qquad (13a)$$

$$RI = y^2$$

An additional constraint is required to assure that physically realistic values of RI are calculated for stable conditions; examination of Equation 12 reveals that a singularity exists for RI = 1/7. The singularity is eliminated by assuming $\phi(RI) = \phi(.137)$ for RI $\geq$ .137. This constraint is implemented only in rare instances during extremely stable conditions. This problem is also evident in Golder's nomogram (17) for estimating RI from B; the RI scale on the nomogram has a maximum value of ~.13.

For unstable conditions Equation 10 can be written as

$$\frac{1-x^4}{16x^2\left[\ln z/z_o+.50864-2[\ln(1+x)]-\ln(1+x^2)+2\tan^{-1}x\right]^2} - B = 0 \qquad (14)$$

where x is given by Equation 2a. Equation 14 is solved by Newton's method.

The methodology for calculation of SIGAR can be summarized as follows:

- Calculate B from available tower or Rawinsonde data (Equation 9)
- Evaluate F(B) (Equation 7)
- Specify z_o. A reasonable first approximation is z_o = .25m (18)
- Solve Equation 13 or 14 (unstable or stable conditions) to obtain RI from Equation 11 or 13a respectively.
- Calculate SIGAR from Equation 1.

Preliminary calculations using Rawinsonde data and data constructed for the purpose of comparison with Cramer Co. SIGAR values (19) are given in Tables 2-8 and 2-9 respectively.

Table 2-6. Calculations of SIGAR using Rawinsonde data between the surface and the first standard pressure level (1,000 mb) with z_o = .25m.

Date ('69, 12Z, 0700 EST)	SIGAR (deg)	$\Delta\theta/\Delta z$ (°C/100m)	U (m/sec)
1/1	10.8	-.075	13.
1/15	7.2	.98	8.
1/29	7.1	.93	11.
2/5	7.7	2.38	3.5
2/12	8.0	1.29	4.
('69, 00Z, 1900 EST)			
7/7	16.3	-.69	6.
7/14	17.2	-.88	6.
7/21	7.6	.83	4.
7/28	15.7	-.20	2.
8/4	14.5	-.90	14.

Table 2-7. Comparison of calculated SIGAR[2] with values given by Dumbauld et. al (19) (in parenthesis)

Wind Speed at 18m (m sec^{-1})	STABILITY CLASS				
	Very Unstable (ΔT=-1.4C)[1]	Slightly Unstable (ΔT = -0.8C)	Near-Neutral or Transitional (ΔT=0C)	Slightly Stable (ΔT = 0.4C)	Very Stable (ΔT=2.0C)
1-2	21.29 (25.)	8.73 (14.)	9.35 (8.5)	8.83 (7.5)	5.4 (5.5)
2-4	18.13 (16.)	10.97 (12.)	8.95 (8.5)	9.10 (7.5)	9.38 (5.5)
4-7	20.03 (12)	11.87 (10)	8.76 (8.5)	8.78 (8.0)	8.95 (7.0)
7-11	15.65 (10)	12.11 (9.5)	8.73 (8.5)	8.73 (8.5)	8.76 [Note 3]

(1) ΔT measured between 3m and 60m.

(2) SIGAR is the standard deviation of the wind azimuth angle measured over a 10-minute period.

(3) Very stable conditions cannot occur with KSC with such large wind speeds.

Testing of the computer code for SIGAR using the 1965 KSC Rawinsonde data has begun. Preliminary results for January and February data are summarized in Tables 2-8 and 2-9. Table 2-8 indicates that none of the computed values of SIGAR were less than 3 degrees, very few were greater than 18 degrees and most were between 6 and 9 degrees. Table 2-9 indicates that for a particular potential temperature gradient, SIGAR increases with decreasing wind speed. Table 2-9 should be expanded to cover more wind speed and potential temperature gradient intervals as the calculations based on all the 1965 data become available.

Table 2-8. Distribution of SIGAR computed from Rawinsonde Data (January, February 1965, 239 soundings)

SIGAR (deg)	Percent Occurrence
<3	0
3-6	22.6
6-9	48.1
9-18	25.5
>18	3.8

Table 2-9. Mean SIGAR (deg) as a Function of Potential Temperature Gradient for Two Wind Speed Intervals (January 1965, 26 cases)

		$\Delta\theta/\Delta Z$ (°c/m)		
U(m/sec)	≤.0017	-.0017 to .0016	.0016 to .0070	.0017 to .0187
4-8	13.6	6.3	6.1	5.9
2-4	17.5	*	*	7.7

*No Data

2.5 MODIFICATION TC THE UNIVAC 1108 VERSION OF THE REED DESCRIPTION AND THE CLOUD RISE PROGRAM

In the area of climatological assessment, one of the major tools is the NASA/MSFC REED Descrip' ›n (2). In the original mode of operation, a pre-processor program was required to read the meteorological data and calculate cloud rise and cloud location. This process has been automated so that the two programs are executed in one job stream. Instead of punching cards, the cloud rise program builds a disk file where each case processed is given a unique identifer. The REED program then executes with the capability of choosing any of the cases from the cloud rise file in any order. Additional flexibility is achieved by allowing the user to override any parameters set by the cloud rise program prior to the execution of the REED Description.

For the purposes of documentation and compact storage, the capability to produce a duplicate copy of all printer output on plot paper was added. This plotter output is much more suitable than printer output for 8½ by 11" documents, and is also more easily filed. In addition, the tapes from which these plots are made can be saved and used as data files from which additional calculations can be performed.

Finally, the capability to print a table summarizing the most critical parameters for each case in a particular run was added to the code. Thus in a run where many cases are processed, the user can quickly determine which cases are the more critical. This table can be conveniently used directly for documentation purposes.

2.5.1 Screening Program Modification

Modifications to provide additional capabilities for the Meteorological Data Screening Program* were completed. The MET Screening reads in cards, which indicate which soundings to search for, and reads from met data tapes, then it generates as output, plots, cards, and printout for each sounding processed.

The plots include a list of the cloud rise heights plus the following plots:

- Wind Speed versus Altitude
- Wind Direction versus Altitude
- Dry Bulb and Potential Temperature versus Altitude
- Temperature, Virtual Temperature, and Virtual Potential Temperature versus Altitude

The card output from the Screening Program is punched in the format needed for the pre-processor.

The printout has been expanded to include stability criteria. The data were tested against both TVA and AEC stability criteria. The results are printed in a table after the original output has been completed for each time. The following information is printed: The altitude interval, temperature interval, DTODS, AEC stability, potential temperature interval, DPTODZ, and TVA stability. DTODZ is defined as

$$\frac{T_i - T_{i-1}}{Z_i - Z_{i-1}}$$

where i = 1 - no. of altitudes
T is temperature (^{o}C)
Z is altitude (meters)

DPTODZ is defined as

$$\frac{PT_i - PT_{i-1}}{Z_i - Z_{i-1}}$$

PT is potential temperature

*Initially developed by Dr. Stephens and W. C. Campbell at MSFC.

The AEC and TVA stability criteria is listed in Table 2-10. This added printout aids in choosing the height of the surface transport layers needed for input into the cloud rise program.

Table 2-10. Stability Criteria

ATOMIC ENERGY COMMISSION CRITERIA

X = Gradient of Temperature

Classification		X(^{o}C/Meter)
Extremely unstable		X< -.019
Moderately unstable	-.019 ≤	X< -.017
Slightly unstable	-.017 ≤	X< -.015
Neutral	-.015 ≤	X< -.005
Slightly stable	-.005 ≤	X< .015
Moderately stable	.015 ≤	X< .040
Extremely stable	.040 ≤	X

TENNESSEE VALLEY AUTHORITY CRITERIA

Y = Gradient of Potential Temperature

Classification		Y(^{o}C/Meter)
Unstable		≤ -.0017
Neutral	-.0017	Y≤ .0016
Moderately stable	.0016	Y≤ .0070
Very stable	.0070	Y≤ .0187
Extremely stable	.0187	<Y

2.5.2 Addition of New Vehicle and Updating of Constants

The characteristics of the newest solid motor in the Thor-Delta family of launch vehicles were added to the Multilayer/pre-processor system. This new vehicle is known as the Thor-Delta 3914 and is the fifth vehicle that can be simulated by the code.

In conjunction with determining the values for constants associated with the 3914, the same constants were examined for the other four vehicles. These constants include the following:

- QC1, QC2, QC3 - total source output rates (g/sec) for the three types of launch respectively (i.e. normal, abnormal with one motor burning on the pad, abnormal where motors explode and burn on the ground).
- QT1, QT2, QT3 - total source strength (g) for the three types of launch respectively.
- HEATN, HEATM, HEATA - Heat output (cal/g) for the three types of launch respectively.
- a, b, c - Rocket rise parameters in the equation $T = az^b + c$ where T is the burn time and z is the altitude
- FRQ1 - Fractional distribution of material for HCl, CO, CO_2 and AL_2O_3.

Table 2-11 lists the preliminary values determined for these constants:

Table 2-11. Preprocessor Program Constants

	Vehicle				
Parameter	Titan III C	Space Shuttle	Thor-Delta 2914	Minuteman II	Thor-Delta 3914
QC1	5.437528E6	1.5219230E7	8.360685E5	4.684476E5	1.057557E6
QC2	2.718764E6	6.882968E6	9.09811E4	4.684476E5	1.4829227E5
QC3	1.359382E6	3.441484E6	2.729434E5	1.171119E5	3.70731E5
QT1	3.262517E8	1.894794173E9	2.887598E7	2.810686E7	6.701631E7
QT2	1.631258E8	8.569295E8	3.14229E6	2.810686E7	9.398616E6
QT3	3.262517E8	1.713859E9	1.885373F7	2.810686E7	4.699308E7
HEATN	2021.1	(1969.6)* 1479.7	1766.0	2055.9	1449.9
HEATM	1010.55	1062.35	1000.00	2055.9	1000.00
HEATA	1000.00	1000.00	690.0	1000.00	411.18
FRQ1					
-HCl	.1931	.1782	.1218	.1977	.1589
-CO	.2665	.2021	.2055	.2380	.2783
$-CO_2$	.0222	.0286	.0156	.0318	.0331
$-AL_2O_3$	.2819	.2524	.2214	.2761	.1936
AA	.429580	.652213	.922156	.469982	1.245756
BB	.518422	.468085	.432703	.463333	.418095
CC	0.375	5.0	0.0	0.0	0.0

*Value used in report; other is up-dated reflecting latest result.

The original rocket rise equation $T = az^b$ was modified to the form $T = az^b+c$. The constant c was added to take into account the time lapse between ignition and lift-off. The parameters a, b, and c were obtained from least squares fits of empirical trajectory data. Plots of the trajectories generated by the old values and by the new values were made against the measured trajectories. The results are shown in Figures 2-7 through 2-11. For each of the original four vehicles, the trajectory generated by the new values is closer to the measurement than is the old trajectory.

Since the burn rate for solid propellant motors is influenced by the initial temperature of the propellent, the pre-processor program was modified to take into account this initial propellant temperature. A table of the mean temperatures at KSC for each month was added to the code. Based on the month in which meteorological data was taken, the default temperature is obtained from the table and used to compute a burn-rate factor (where 70° is the standard, yielding a burn-rate factor of one). The capability to over-ride this default table was also added to the code, so the initial propellant temperature, if known, can be input to the program.

Various runs were made with the UNIVAC 1108 REED Description/Cloud Rise system to check out all the modifications made to the code; however no production type runs have been performed.

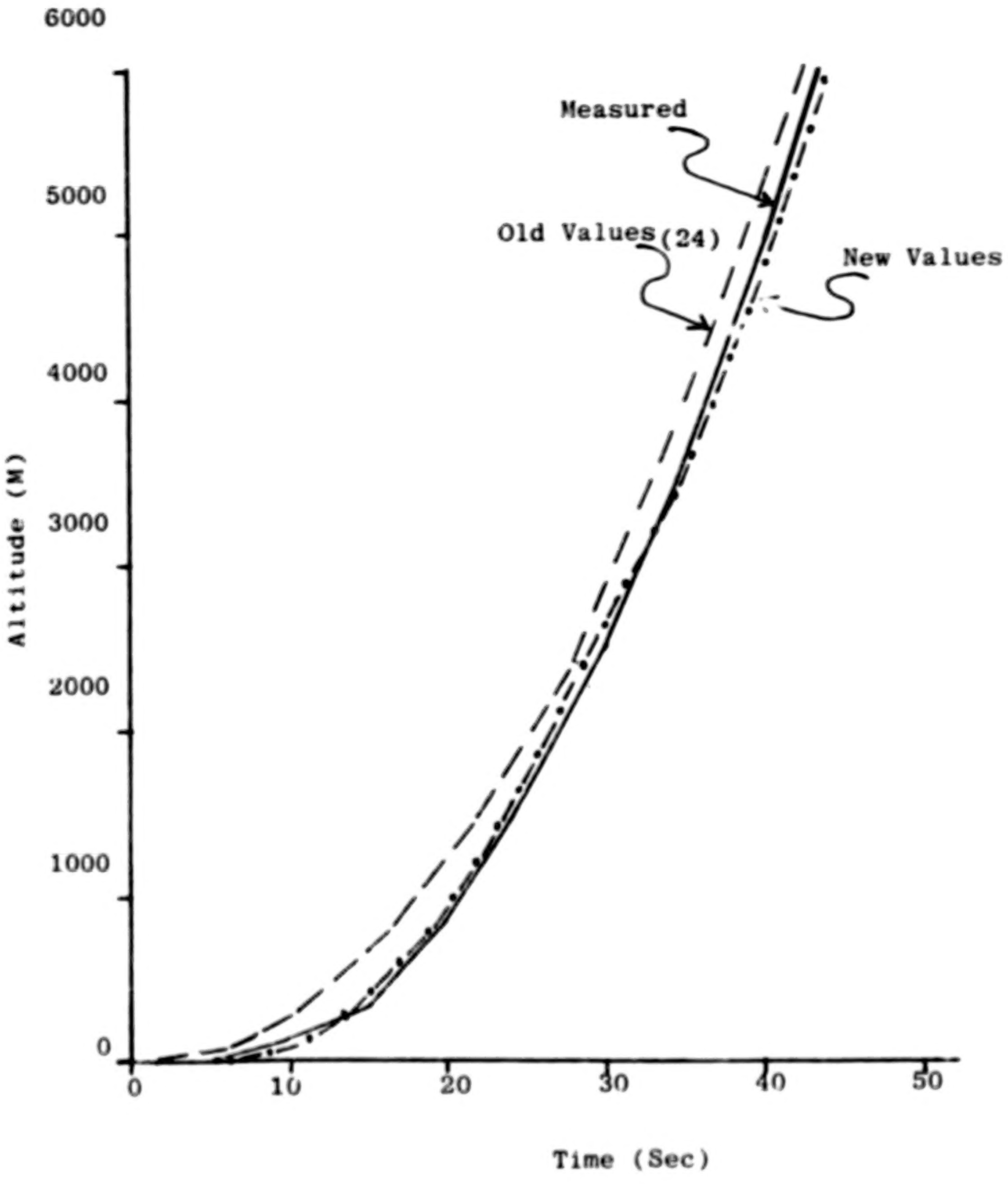


Figure 2-7. Titan III C Trajectory
$T = az^b+c$

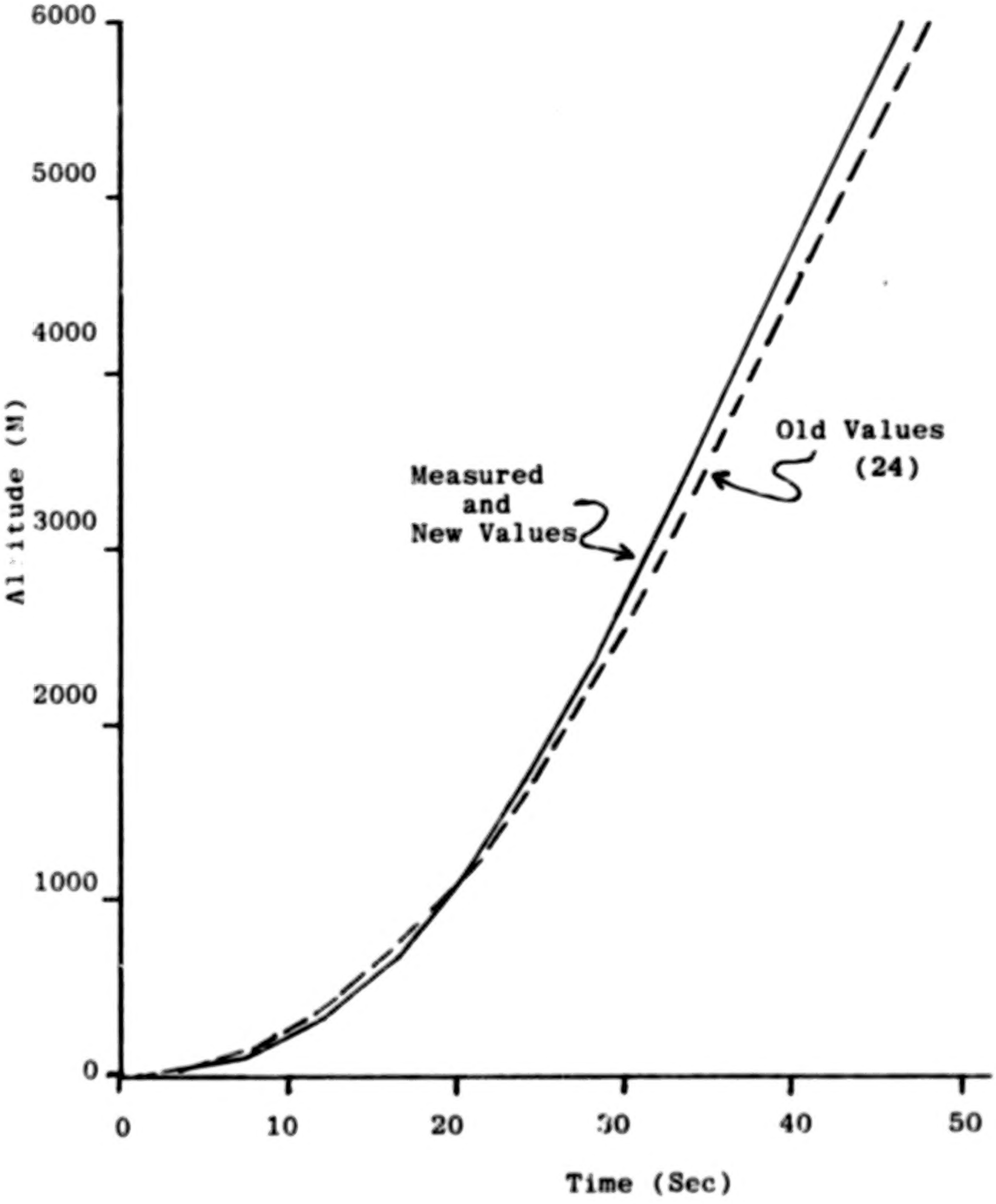


Figure 2-8. Space Shuttle Trajectory
$T = az^b+c$

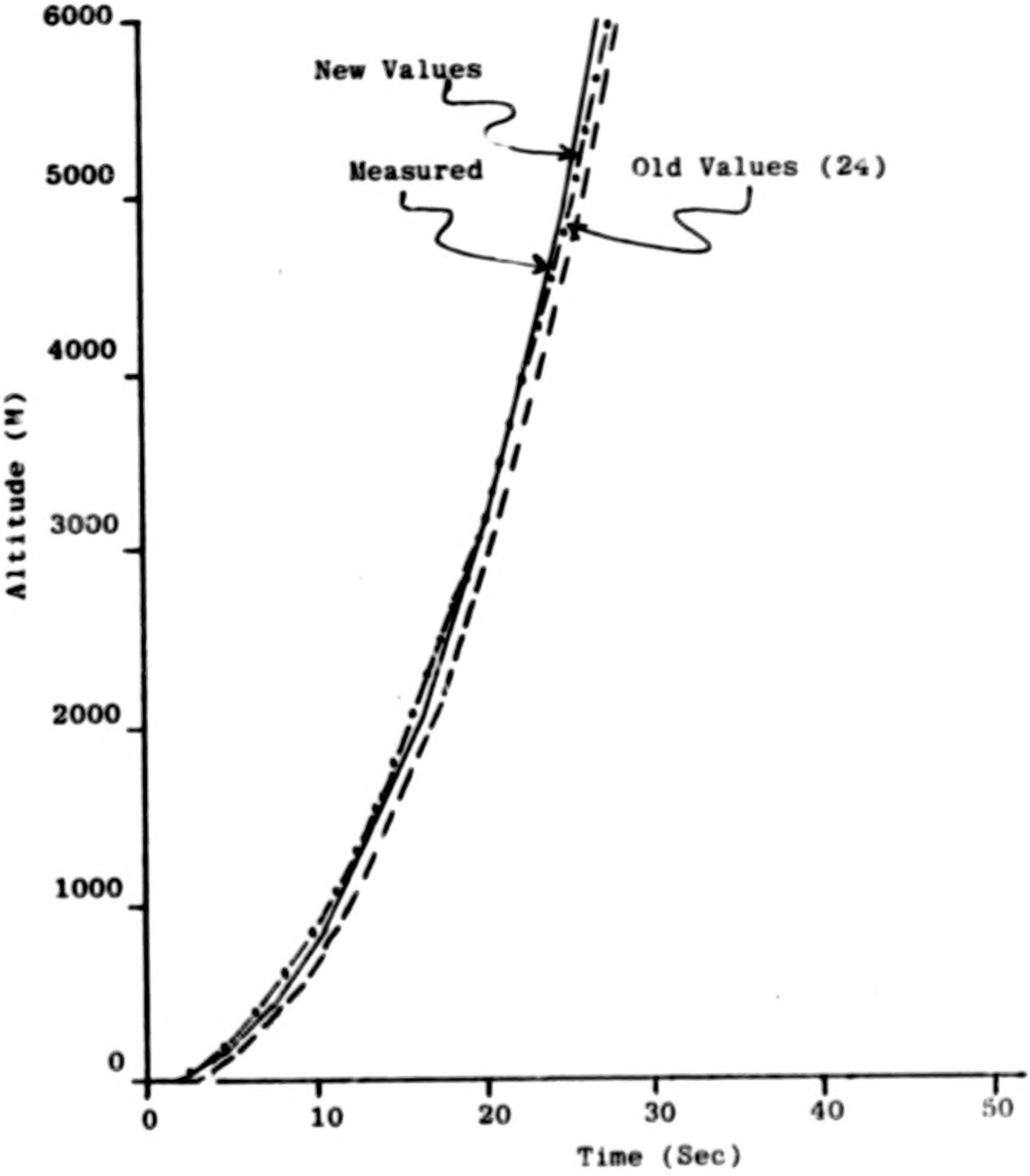


Figure 2-9. Minuteman II Trajectory
$T = az^b+c$

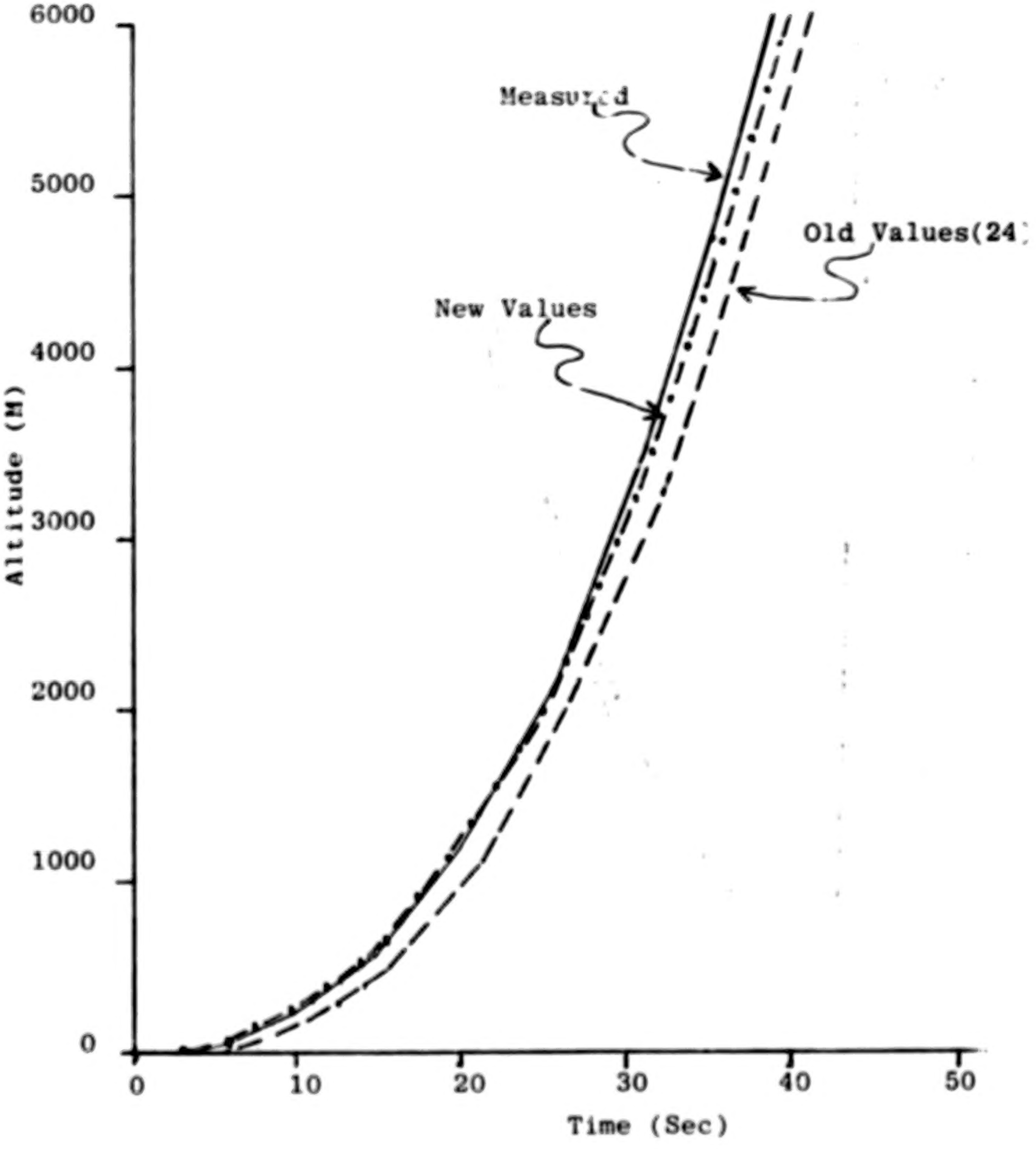


Figure 2-10. Delta Thor 2914 Trajectory
$T = az^b+c$

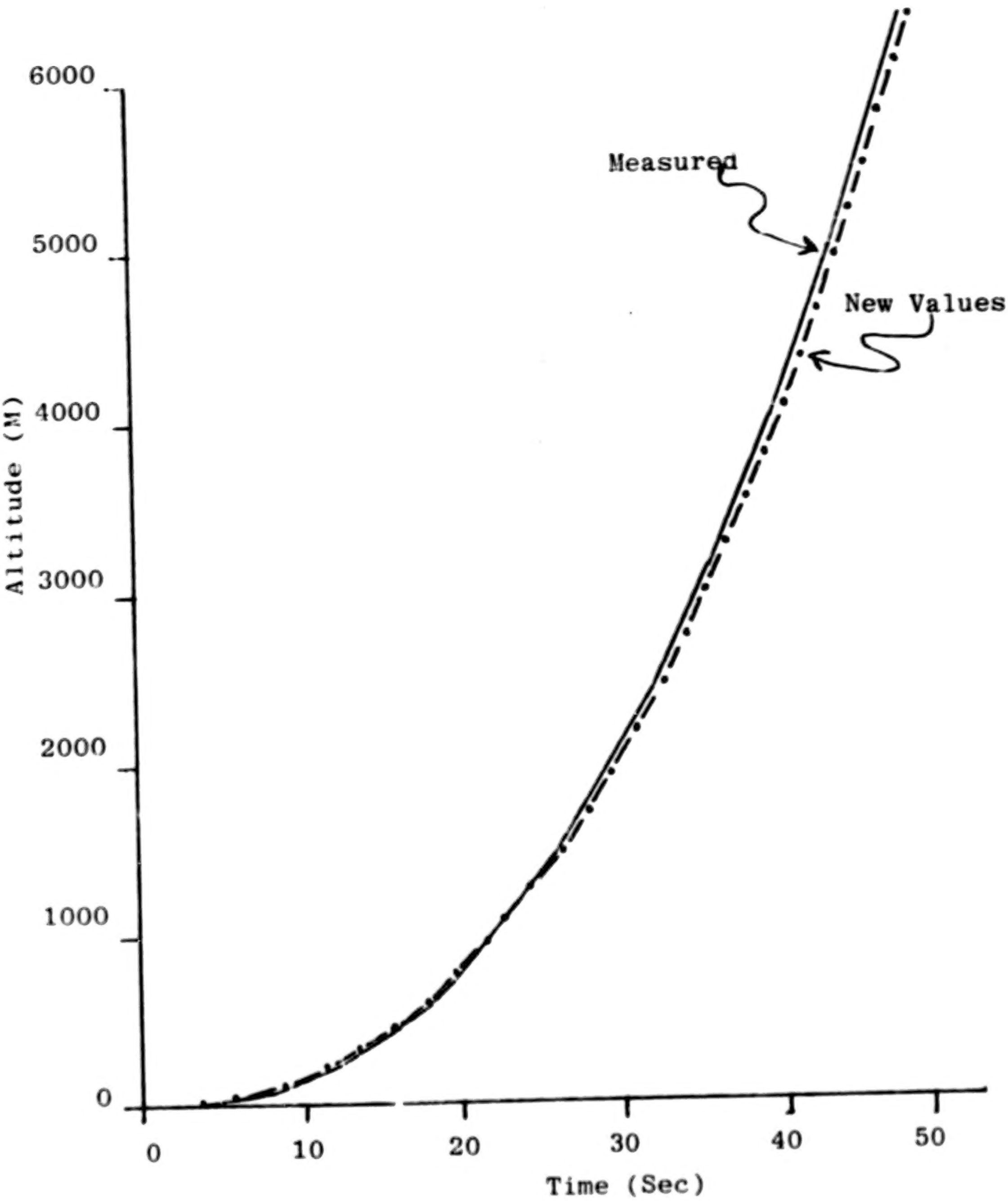


Figure 2-11. Delta Thor 3914 Trajectory(21)
$T = az^b+c$

2.6 SURFACE DEPOSITION MODEL

Three major constituents in the SRB and SSME exhaust effluents are Al_2O_3, HCl, and H_2O. The HCl is of prime concern environmentally since it is potentially toxic in the gaseous phase and forms a strong acid in the aqueous phase. Two phases of Al_2O_3 can exist, the gamma phase which reacts strongly with HCl and the alpha phase which does not. The HCl reacts with and is absorbed by water droplets to form an aerosol. The formation of the aerosol, of course, reduces the concentration of gaseous HCl in the atmosphere. The aluminum oxide absorbs H_2O readily; it is used as a drying agent in laboratories. Rain falling through a cloud consisting of the rocket exhaust effluents and the entrained air can react chemically with the HCl and possibly the Al_2O_3 and can physically interact with the HCl aerosol and the Al_2O_3. Thus it can be seen that the $Al_2O_3/HCl/H_2O$ system has a large number of physical and chemical interactions that can occur simultaneously. A consistent set of reactions and interactions must be developed to allow the calculation of the HCl and Al_2O_3 concentrations for a surface deposition model.

The phase of the aluminum oxide in the exhaust is not without question. Early work(25) indicated that the aluminum oxide present in a rocket exhaust is the alpha phase, which does not react with HCl. More recent data(26,27,28) has indicated that some of the gamma phase may be present. This may be important to the surface deposition depending on the aluminum particle size distribution in the rocket exhaust. The SRB exhaust will have relatively large particles; therefore, the amount of gamma phase will be less than found in small motor firings. Of course, for an equal weight, the number and surface area for small particles is greater than for larger particles. The whole question of Al_2O_3 phase is undergoing intensive investigation and must be considered when developing a surface deposition model.

Using available experimentally measured Al_2O_3 particle size distributions (29) for solid propellant rocket motors of various sizes and making reasonable assumptions as to particle size growth as a function of throat diameter, a particle size distribution for the Al_2O_3 exhausting from the Space Shuttle SRB was determined. This is shown in Table 2-12. At the present time no realistic input to the REED Description surface deposition model is available for use; therefore, these effects, which may be significant for climatological predictions, have been neglected.

2.7 ABSORPTION AND SCAVENGING

Studies have been conducted on atmospheric scavenging of HCl which experimentally determined the washout coefficient (30,31). Washout involves several microprocesses, including the solubility of HCl in raindrops, the diffusion of HCl to the falling raindrops, and the physical parameters which characterize the rain. At higher relative humidities, washout of HCl aerosol must be considered in addition to the washout of gaseous HCl. The Al_2O_3 particles as well as salt or dust particles in the rocket exhaust may act as potential cloud droplet nuclei. The nucleating efficiency of Al_2O_3 particles is unknown at this time. The rain scavenging experimental results must be integrated into the surface deposition model.

The effects of absorption and scavenging, which may be significant for climatological predictions, have been neglected in this study because of the lack of a suitable, acceptable washout coefficient. (27,30,32,33)

Table 2-12

SRB Particle Size Distribution[29]

Particle Diameter in Microns	Weight Percentage of the Particles of that Size Range
0- 7.0	20.0
7.0-10.0	20.0
10.0-14.0	20.0
14.0-16.0	20.0
16.0-23.0	20.0

2.8 CONCLUSIONS

The long term objective of this study is to establish the relation between weather patterns of various scale and the environmental impact of the Space Shuttle exhaust effluents. To date, the synoptic weather patterns have been categorized and their relative frequency of occurrence have been calculated.

Concurrently the tools for calculating air quality assements for large samples of KSC meteorological data have been developed. A large sample of Rawinsonde data are available for definition of the variability of calculated air quality assessments over time scales as small as six hours. This variability is associated with such phenomena as the development of the sea breeze and ground based stable layers. These phenomena strongly influence the critical meteorological input variables to the diffusion model.

3. EXHAUST CHEMISTRY

The calculation of the heat content of the plume, or more exactly the heat content of the rocket exhaust effluents, taking into account interactions with the ambient environment, is a well-defined problem. The problem has been attacked for many years by the propulsion community and a set of standard techniques have been devised and published for liquid engine performance and analysis by the Interagency Chemical Rocket Propulsion Group - Joint Army, Navy, NASA, Air Force, (ICRPG-JANNAF) Performance Standardization Working Group (34, 35, 36). The state-of-the-art of analysis for solid motors is not yet as advanced but an ICRPG-JANNAF Solid Performance Working Group has begun work.

The available techniques were adequate for analyzing the plume from the liquid propellent SSME rocket engine and the solid propellant SRB motors. The value of the effective heat release and the exhaust species concentrations were quantitatively satisfactory for both propulsion devices. During this study only the Space Shuttle SRB exhaust effluents were studied in detail. Solid propellant rocket motors have the phenomena of two-phase flow occurring in the combustion chamber, nozzle, and plume. The two phases are not in thermal or velocity equilibrium. In general, the particles, in this case solid and liquid aluminum oxide, are traveling slower than the gas, are at a higher temperature than the gas, and are at a greater flow angle than the gas. These phenomena make the characteristics of a two-phase flow field different than that of a single-phase flow field such as occurs in the liquid propellant SSME.

3.1 TWO-PHASE FLOW PHENOMENA

The analysis of the two-phase flow in the SRB rocket nozzle started with a one-dimensional thermochemical analysis of the solid propellant. As seen in Figure 3-1, when solid propellant combusts, the combustion products are at some pressure, P_c, and some flame temperature, T_c. The chamber pressure history is governed principally by the amount of burning surface exposed. The desired amount of burning surface (chamber pressure) can be obtained by the geometry of the propellant grain. Figure 3-2 shows the Space Shuttle altitude, Mach number, and Solid Rocket Motor chamber pressure history for the first 70 seconds of flight. As can be seen, the chamber pressure varies from 825 to 580 psia during this portion of the flight. With a knowledge of the propellant composition and the chamber pressure, the flame temperature and the concentrations of the combustion products were calculated as shown in Table 3-1. The flame temperature and the combustion products as a function of time (velocity and altitude are then known) were needed for input to subsequent steps. The calculations were performed on the NASA UNIVAC 1108 with a program written by NASA-Lewis Research Center (38) and modified by SAI.

By means of two-phase characteristic theory, the supersonic portion of the flow field of the SRB nozzle and plume was determined. With reference to Figure 3-1, the supersonic portion is bounded roughly upstream by the nozzle throat and downstream by the plume boundary. The nozzle analysis portion of the program basically terminates calculation along the last left-running characteristic, identified on Figure 3-1. This surface is significant in that no disturbance downstream of it will affect the pressure field along the nozzle wall. The program originally written by TRW personnel (38) and extensively modified by SAI (40) yields vital pieces of information along the last

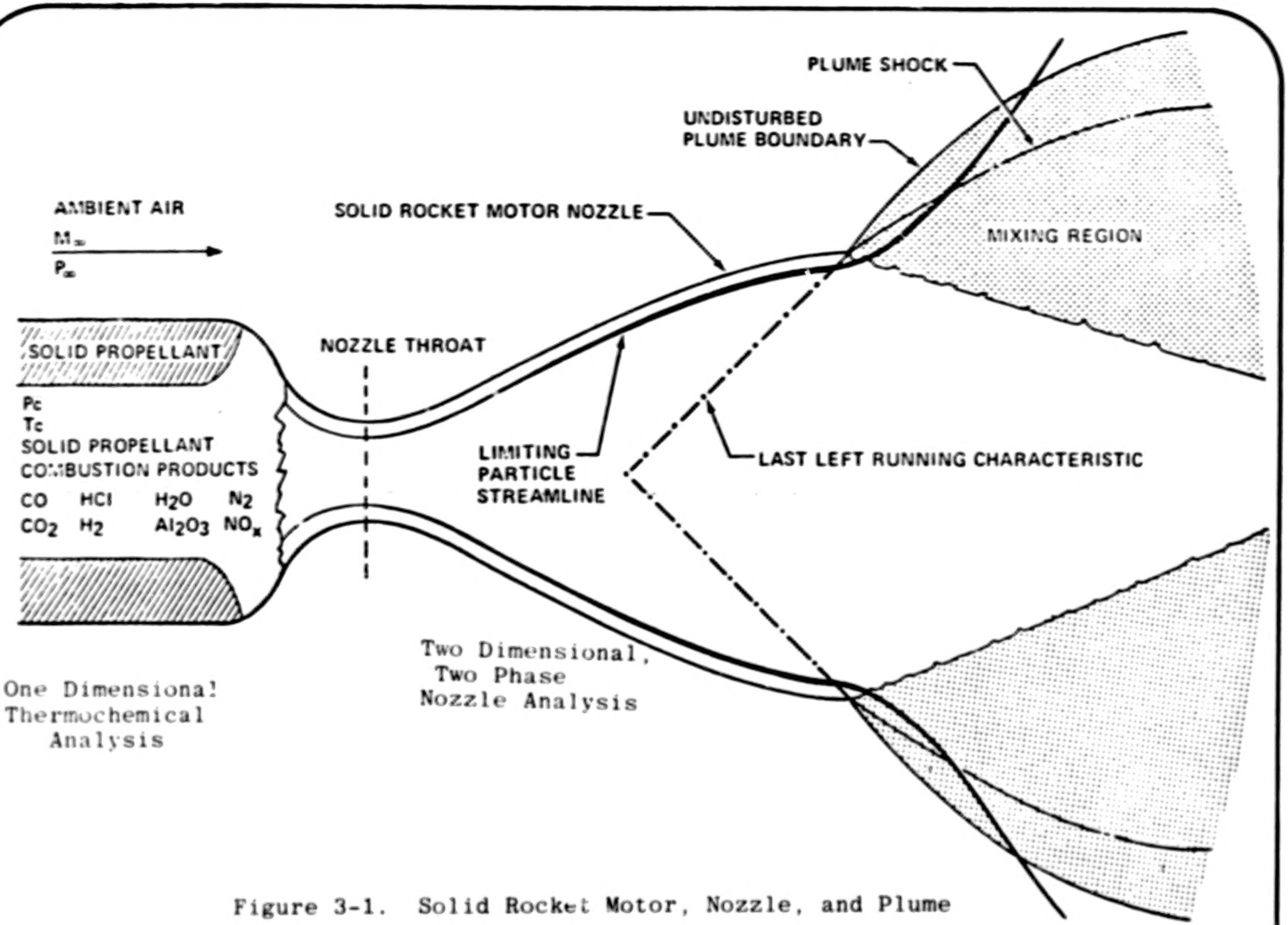

PLUME SHOCK
UNDISTURBED PLUME BOUNDARY
AMBIENT AIR
M_∞
P_∞
SOLID ROCKET MOTOR NOZZLE
MIXING REGION
SOLID PROPELLANT
NOZZLE THROAT
P_c
T_c
SOLID PROPELLANT COMBUSTION PRODUCTS
CO HCl H_2O N_2
CO_2 H_2 Al_2O_3 NO_x
LIMITING PARTICLE STREAMLINE
LAST LEFT RUNNING CHARACTERISTIC
Two Dimensional, Two Phase Nozzle Analysis
One Dimensional Thermochemical Analysis

Figure 3-1. Solid Rocket Motor, Nozzle, and Plume

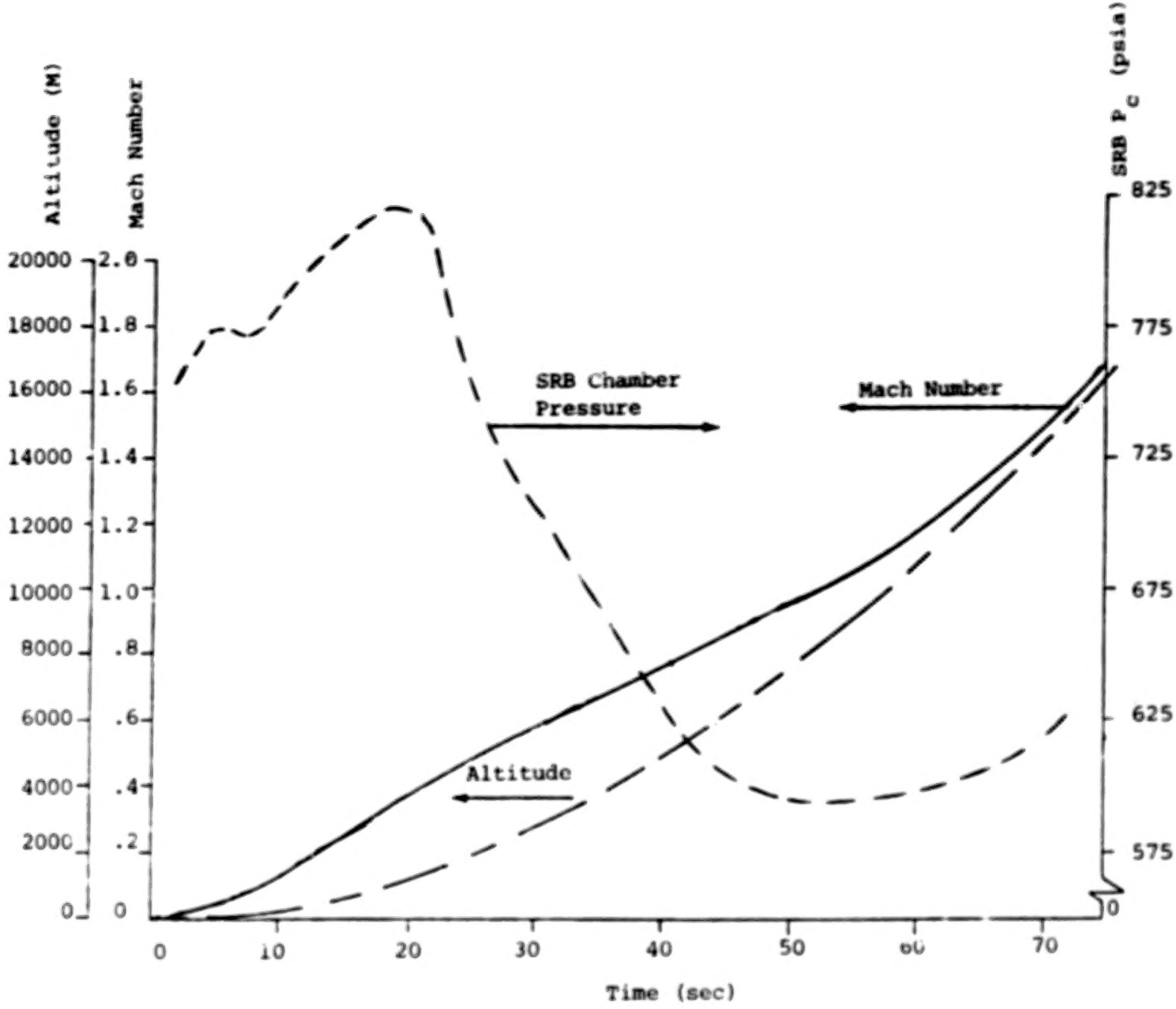


Figure 3-2. Shuttle Altitude, Mach Number, and SRB Chamber Pressure History

Table 3-1

Space Shuttle SRB Exhaust Effluents at Chamber, Throat and Exit Plane

Motor Conditions

P_c = 780 psia
T_c = 3426°K
A_e/A_t = 7.15

Species	Location in Motor		
	Chamber	Throat	Exit Plane
	Concentrations in Mole Fractions		
AlCl	0.0040	0.0025	0.0000
$AlCl_2$	0.0016	0.0011	0.0000
$AlCl_3$	0.0002	0.0001	0.0000
AlOCl	0.0015	0.0010	0.0000
AlOH	0.0004	0.0002	0.0000
AlO_2H	0.0006	0.0003	0.0000
Al_2O_3	0.0738	0.0758	0.0798
CO	0.2316	0.2326	0.2326
CO_2	0.0160	0.016[illegible]	0.0209
Cl	0.0115	0.0096	0.0019
H	0.0333	0.1[illegible]67	0.0046
HCl	0.1350	0.[illegible]411	0.1575
H_2	0.2586	0.[illegible]630	0.2791
H_2O	0.1405	0.1406	0.1391
NO	0.0006	0.0004	0.0000
N_2	0.0818	0.0824	0.0841
O	0.0006	0.0004	0.0000
OH	0.0081	0.0056	0.0004
O_2	0.0003	0.0002	0.0000

left-running characteristic which are needed for subsequent steps in the modeling.

A two-phase flow field is dissipative and non-equilibrium in nature. There is, therefore, an entropy rise down the flow field and an entropy gradient radially across the flow fields since the particles and gas have a different history at every point in the flow field. The entropy rise and the loss in total pressure can be calculated from local flow properties. Figure 3-3 shows the total pressure loss and gradient along the last left-running characteristic for the Space Shuttle SRB nozzle with a single particle size of 12.0 micron diameter which represents an average particle size. The pressure loss varies from about 27 to 55 percent of the chamber pressure; thus, the species and energy content of the exhaust will vary across the nozzle exit. Because of the wide variation in properties across the exit, an integration scheme was incorporated into the program which integrates the mass flow and energy content and computes the average for a gross value of the energy content of the exhaust as it leaves the nozzle. The energy content of the exhaust was assumed to be composed of two parts: the sensible enthalpy and the kinetic energy of the gas. For a SRB operating at 780 psia chamber pressure, the average integrated value of the heat content of the plume is 2125 calories per gram. Figure 3-4 is a schematic of the SRB nozzle. The chamber pressure chosen, 780 psia, is an average value representative of the SRB when it is close to the launch pad, 0-3000 meters altitude.

3.2 AFTERBURNING AND MIXING ANALYSIS

Solid propellants normally are formulated to have an exhaust composition rich in underoxidized species, i.e., the carbon, C, is preferentially in the form of carbon monoxide, CO, rather than carbon dioxide, CO_2. This formulation technique gives higher specific impulse for the propellant. A

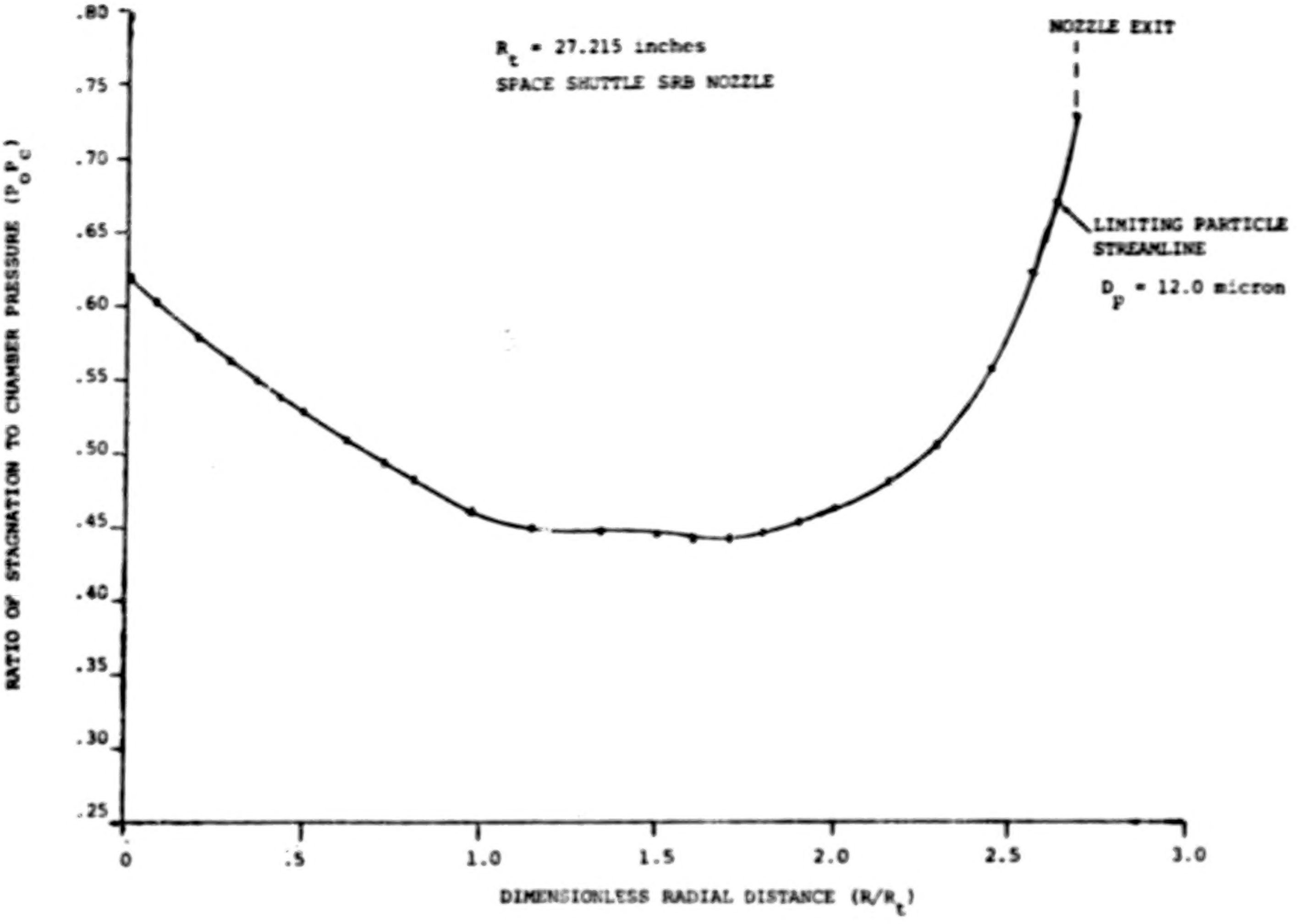


Figure 3-3. Ratio of Stagnation to Chamber Pressure vs Radial Distance Along Last Left-Running Characteristic

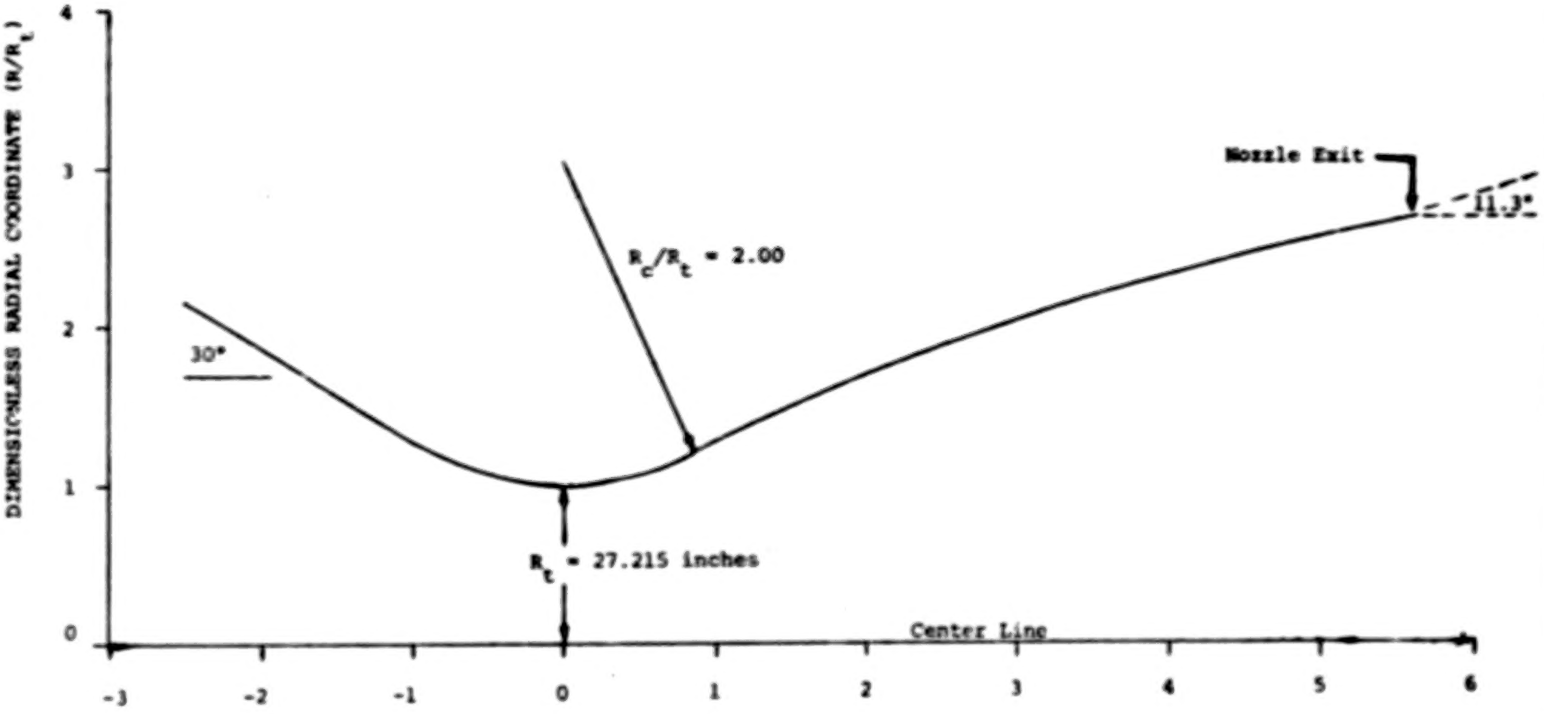


Figure 3-4. DIMENSIONLESS AXIAL COORDINATE (X/R_t)
SRB Nozzle Diagram

jet exhausting into a stationary or moving atmosphere tends to entrain and mix the atmosphere with the jet exhaust.

When a hot exhaust with underoxidized species mixes with ambient air, the possibility for afterburning exists. This condition has been noted on several launch vehicles, and digital computer programs have been written to describe the phenomena with varying degrees of success (41-43). Several of the programs appear generally suitable for use in the proposed study. Based on such factors as ease of input, accuracy, computing time, and calculation technique, the program written by AeroChem Research Laboratories, Inc. (43) was chosen. Figure 3-5 shows the plume afterburning schematic as it applied to this situation. The only modification necessary for the program to be used for the problem under consideration is in the description of the the initial data line. The initial data line for the original program is assumed to be radial, normal to the axis at the exit of the nozzle. All species and the velocity at each grid point must be input. The output of the two-phase analysis program is along the last left-running characteristic. The velocity, entropy, and stagnation pressure are known at every point on the characteristic but not the species; therefore, a technique was devised which would fill the gap between the last left-running characteristic and the needed initial value line and which would calculate the species along the initial value line. There exists in general use in the aerothermodynamic community in this country, a computer program known as PLIMP (44) which calculates and outputs the species concentration, pressure, temperature, and velocity fields on surfaces immersed in a plume; therefore, if a flat plate is placed normal to the axis at the exit of the nozzle, all necessary quantities will be obtained.

The Aerochem mixing program calculated the required values of species concentration and amount of entrained air simultaneously. Stedman (27) in his work estimated the

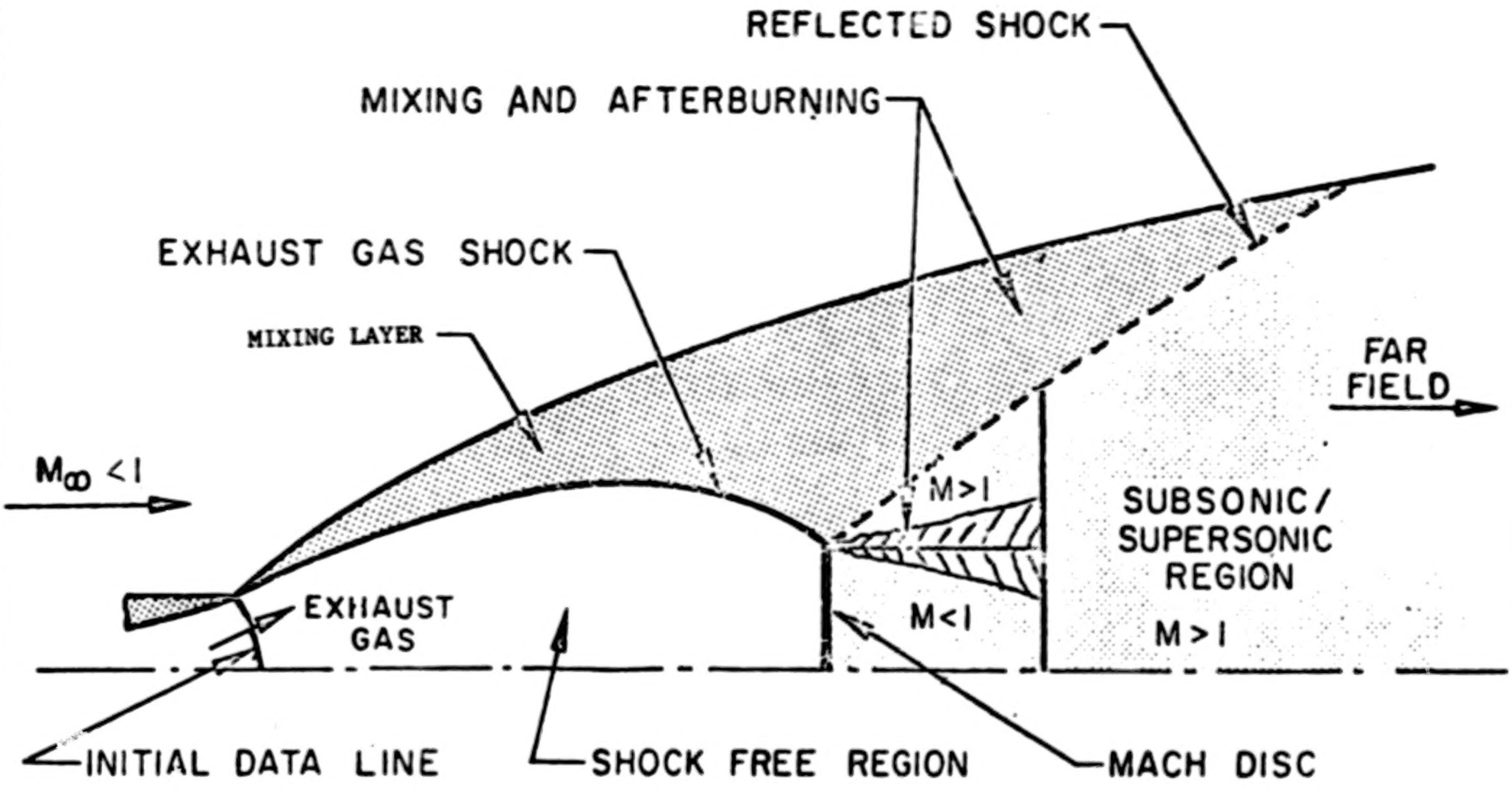


Figure 3-5. Plume Afterburning Schematic

amount of mixed or entrained air from the work of Hart(45) and assumed uniform mixing and chemical equilibrium in the cloud. The Aerochem program not only estimated the amount of entrained air, which is not uniform across the jet, but also determined the species concentrations using finite-rate chemistry. This technique thus detailed the species, the reaction rates, and the temperature and pressure radially across the exhaust as well as in a downstream direction from the nozzle exit. An inventory of the constituents was thus maintained. Table 3-2 lists the reaction scheme utilized in this study. The scheme models the chlorine species production and destruction in detail. Figures 3-6 and 3-7 show the Space Shuttle SRB exhaust effluents as a function of distance from the nozzle exit. Table 3-3 shows the exhaust effluent weight fractions as a function of distance from the nozzle exit.

3.3 OTHER LOSSES

It should be noted a number of potentially important effects were neglected. The first effect neglected was the injection of water into the exhaust. Since the study was initiated, the decision was made to inject large quantities of water into the exhaust as a noise suppression technique. The water is not only to be injected when the Space Shuttle is sitting on the launch pad but the injection will continue until it clears the launch pad. Due to the afterburning and the large number of hot particles, a large luminous plume is formed; therefore, radiation losses may be important. Hart (45) using geometric flight and launch hardware radiant flux estimates, states that the radiation loss may be as large as one-fourth the total heat content. If this estimate is correct, radiation loss calculations are imperative. The radiation data for the exhaust effluents have been collected and tabulated, but the entire calculation has not yet been performed. During the time the SSME's are building up thrust and until shortly after SRB ignition, the Space Shuttle is held onto the launch pad. During this time and even after liftoff, for

Table 3-2

AFTERBURNING ANALYSIS

REACTIONS BEING CONSIDERED

1.	HCL	+	OH		= H20	+	CL
2.	H	+	HCL		= CL	+	H2
3.	OH	+	CL		= HCL	+	O
4.	CL	+	HO2		= HCL	+	O2
5.	CLO	+	OH		= HO2	+	CL
6.	H	+	CL2		= HCL	+	CL
7.	O	+	HCL		= CL	+	OH
8.	CL	+	O3		= CLO	+	O2
9.	CL	+	CL	+ M	= CL2	+	M
10.	O	+	CL	+ M	= CLO	+	M
11.	CLO	+	H		= HCL	+	O
12.	O	+	CLO		= CL	+	O2
13.	H	+	CL	+ M	= HCL	+	M
14.	O3	+	O		= O2	+	O2
15.	O	+	O	+ M	= O2	+	M
16.	O	+	H	+ M	= OH	+	M
17.	H	+	H	+ M	= H2	+	M
18.	OH	+	OH		= H20	+	O
19.	H	+	O2		= OH	+	O
20.	H	+	HO2		= OH	+	OH
21.	H	+	O2	+ M	= HO2	+	M
22.	O	+	H2		= OH	+	H
23.	O	+	HO2		= OH	+	O2
24.	OH	+	HO2		= O2	+	H20
25.	H2	+	HO2		= H20	+	OH
26.	H	+	OH	+ M	= H20	+	M
27.	H	+	HO2		= H2	+	O2
28.	OH	+	H2		= H20	+	H
29.	N	+	O2		= NO	+	O
30.	NO	+	O	+ M	= NO2	+	M
31.	NO	+	CLO		= CL	+	NO2
32.	NO	+	O3		= NO2	+	O2
33.	NO2	+	H		= NO	+	OH
34.	N	+	NO		= N2	+	O
35.	CO	+	OH		= CO2	+	H
36.	CO	+	O	+ M	= CO2	+	M
37.	CO	+	HO2		= CO2	+	OH
38.	NO	+	CLO		= CL	+	NO_2

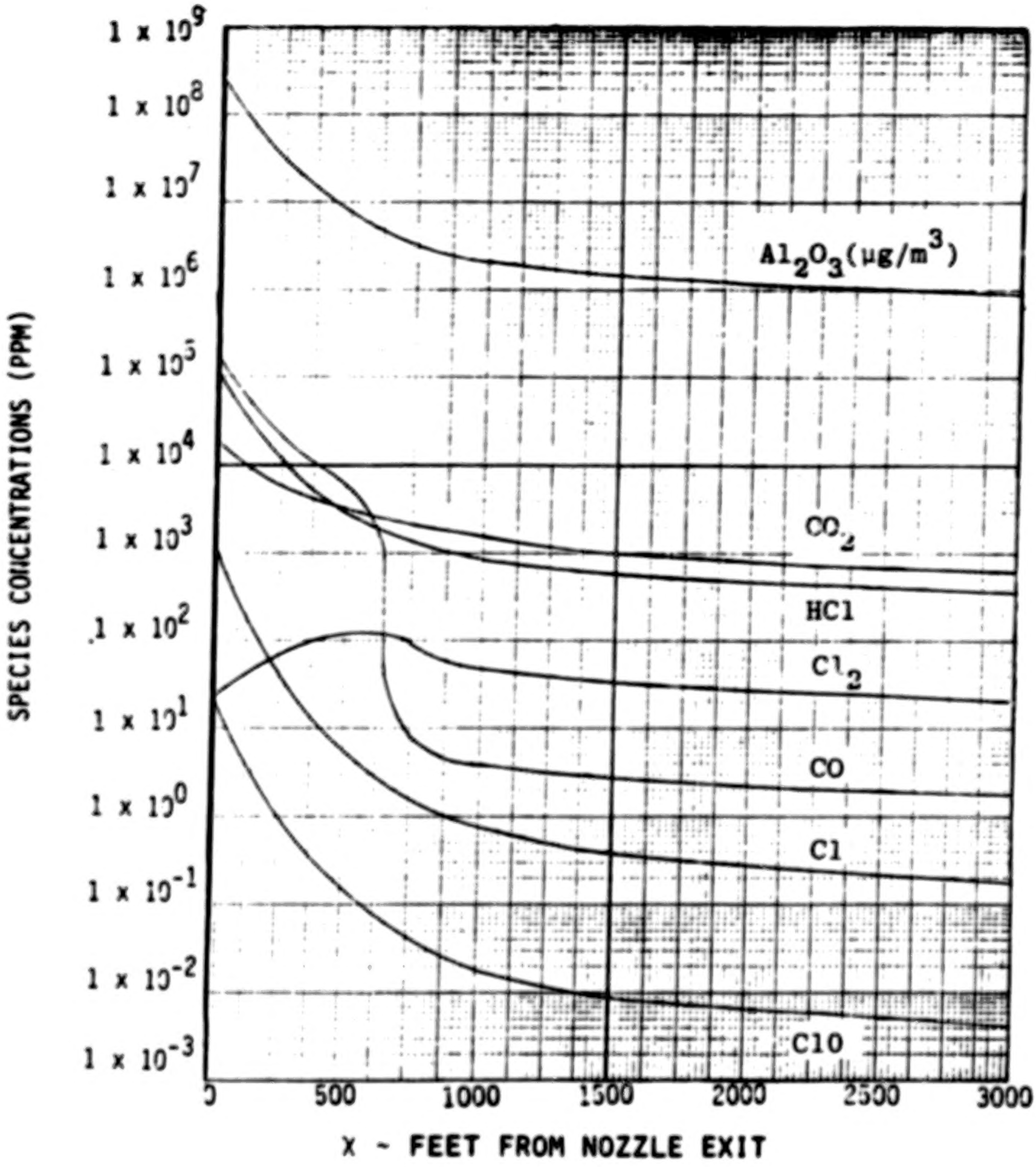


Figure 3-6. Space Shuttle Solid Rocket Motor Exhaust Effluents

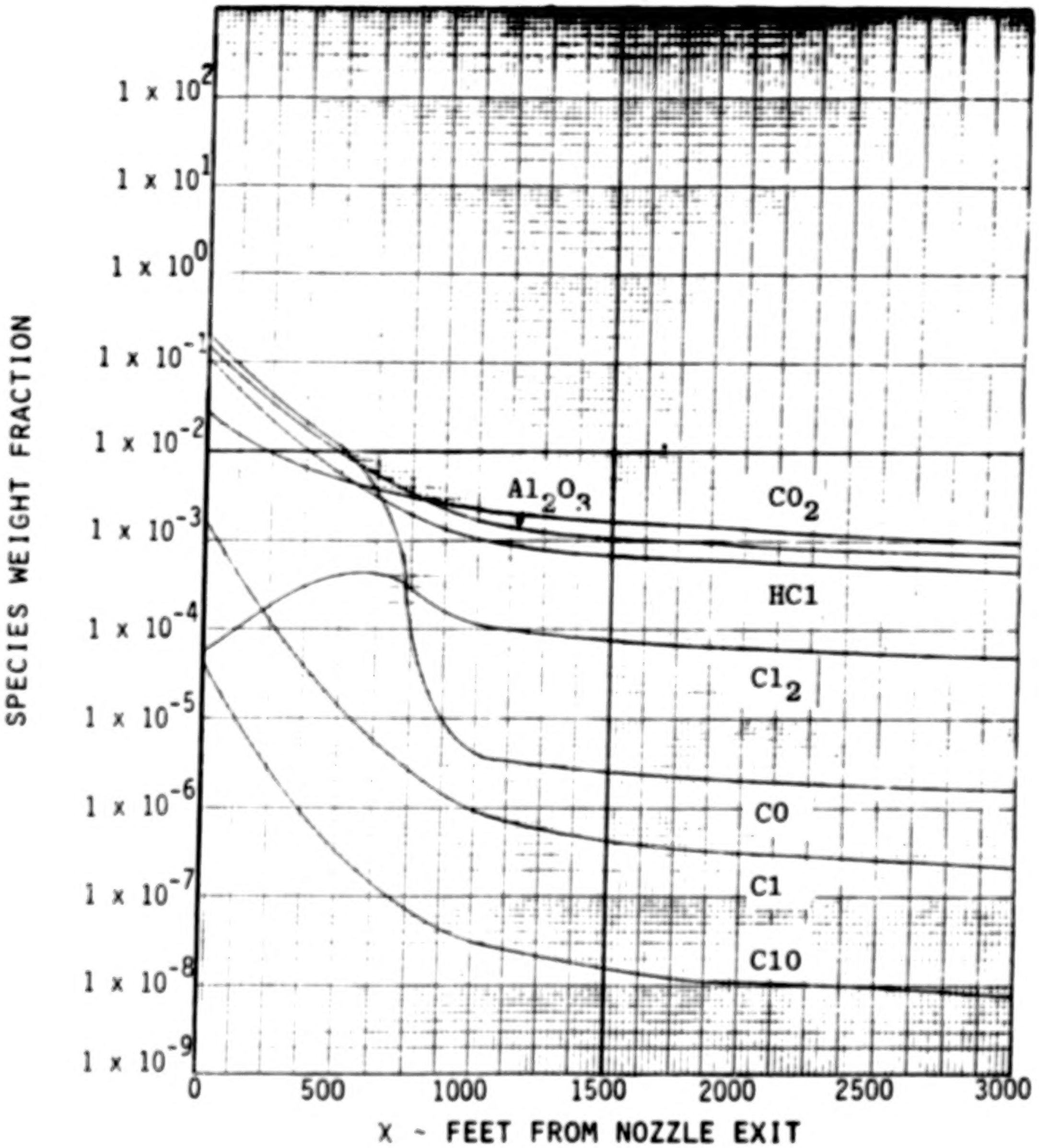


Figure 3-7. Space Shuttle Solid Rocket Motor Exhaust Effluents

Table 3-3

Space Shuttle SRB Exhaust Effluents

Concentrations in Weight Percent

Effluents	Distance From Nozzle Exit - Feet			
	0	1000	2000	3000
Al_2O_3	30.32	22.56	22.36	22.43
CO	24.36	0.052	0.052	0.052
CO_2	3.33	30.85	30.58	30.67
Cl	0.246	0.013	0.008	0.007
ClO	0.006	0.000	0.000	0.000
Cl_2	0.008	1.60	1.59	1.60
HCl	21.41	14.18	14.06	14.10
H_2	2.09	0.000	0.000	0.000
H_2O	9.39	21.43	21.24	21.30
N_2	8.78	8.26	8.13	8.22
NO	0.001	0.989	0.980	0.982
O_2	0.004	0.000	0.000	0.000

a short time, the exhaust effluents are flowing into the flame trench where they are mixed with a large amount of water and ducted away. This mass and energy ducted away represents a possible loss to the ground cloud which may be important. This portion of the problem has not yet been attacked. This study has concentrated primarily on the SRB exhaust effluents and has essentially neglected the SSME exhaust and the problem of the impingement and mixing between the SRB and SSME exhaust plumes. The SSME exhaust effluents have been calculated and are shown in Table 3-4.

3.4 CONCLUSIONS

This study has developed a technique that allows the exhaust effluent chemistry for the SRB to be determined with a state-of-the-art analysis. At this point the basic exhaust effluents have been calculated but a number of important losses such as plume impingement, radiation, flame trench, and water injection have not been addressed. The effluents from the SSME have been determined but the chemical and physical interactions between the two plumes has not been studied.

Table 3-4

SSME Exhaust Effluents

Engine Conditions

P_c = 3000 psia
O/F = 6
A_e/A_t = 77.5

Species	Location in Engine		
	Chamber	Throat	Exit Plane
	Concentrations in Mole Fractions		
H	0.0270	0.0217	0.0000
H_2	0.2477	0.2450	0.2440
H_2O	0.6831	0.7026	0.7560
O	0.0024	0.0015	0.0000
OH	0.0373	0.0276	0.0000
O_2	0.0026	0.0017	0.0000

4. CONVERSION PROGRAMS

In order to allow for the processing of various types of data, received from numerous sources (i.e., Marshall Space Flight Center, Kennedy Space Center, Vandenberg AFB, Point Mugu, Asheville, etc.) and generated on several different computer configurations (i.e., IBM 360, IBM 370, IBM 7094, UNIVAC 1108, CDC 3300 etc.); there becomes a specific need for software which provides the capability of converting the various and voluminous amount of data into the proper BCD/EBCDIC→ASCII character set and record format to make it compatible to the different computer systems (i.e., IBM, UNIVAC, REEDA), upon which the data will be processed by a variety of programs. The data and programs must be available on all NASA/MSFC machines since no machine outage should cause a lack of monitoring capability.

In this section various conversion programs that were developed are discussed. Section 4.1 describes the conversion programs which have generic applications while Section 4.2 discusses conversion programs developed for individual cases. The program listings are given in the Appendix.

4.1 CONVERSION PROGRAMS (GENERIC)

The most efficient means to load data or software on the REEDA System, which was generated on other computer configurations, is to generate a magnetic tape compatible with the REEDA System. The following is a list of the requirements that all magnetic tapes must satisfy to be usable on the REEDA System:

- 9-track magnetic tape
- 800 bits per inch
- Odd parity
- 7-bit ASCII (The 8th bit is always off; this limits the character set to 128 combinations)

however, most (if not all) of the data and software programs being processed on the REEDA System received from other computer installations were recorded on magnetic tape in a format not compatible with the REEDA System. Thus, various conversion programs were generated to convert data recorded on IBM, UNIVAC, CDC, etc., computers to a usable format.

4.1.1 IBM 370/360 BCD → ASCII Conversion Programs

A conversion program was developed to convert data or programs recorded in BCD (on cards or magnetic tape) to a usable REEDA System ASCII character set. This program is written in IBM assembly language and will execute on either the IBM 370 or 360 configuration. It will accept as input either a 7-track or 9-track tape, or punched cards and convert each BCD character into a 7-bit ASCII character compatible with the REEDA System. This converted data is written onto a magnetic tape for REEDA utilization (i.e., 9-track, 800 BPI, ODD parity). An example flow of the conversion process is given in Figure 4-1. Note, only the control cards change when running this program on the IBM 370 or IBM 360.

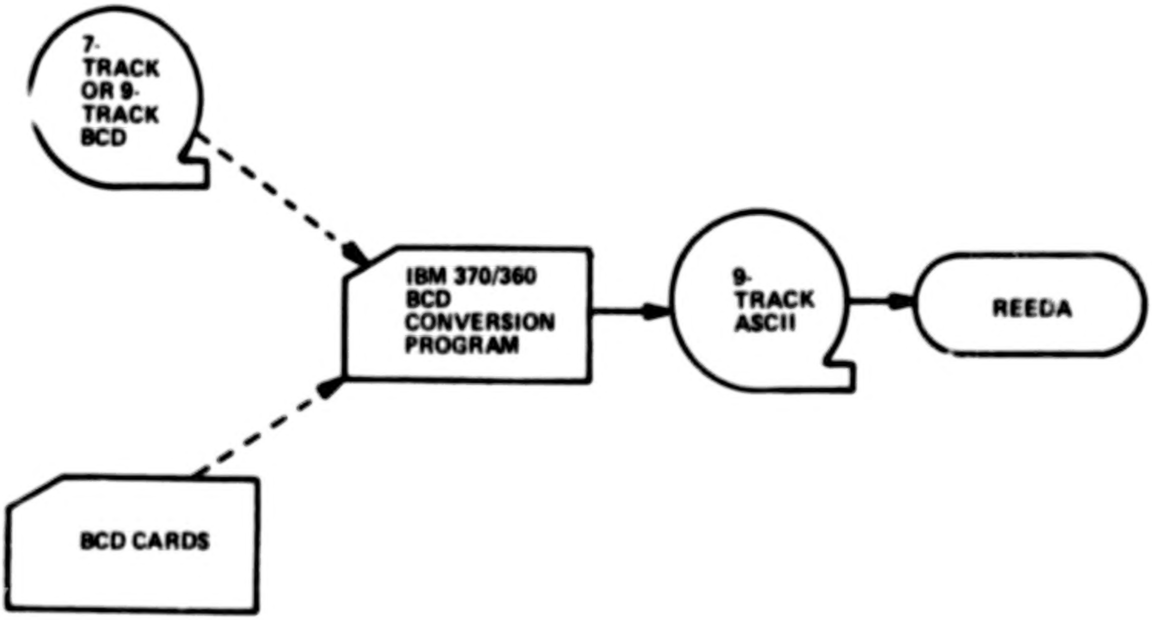


Figure 4-1. IBM 370/360 BCD → ASCII Conversion Process

4.1.2 IBM 370/360 EBCDIC → ASCII Conversion Programs

A conversion program was developed to convert data or programs recorded in EBCDIC (on cards or magnetic tape) to a usable REEDA System ASCII character set. This program is identical to the BCD → ASCII conversion program except all EBCDIC characters are converted to ASCII. As shown in Figure 4-1, input can be either cards or magnetic tape with the output being a REEDA compatible 9-track ASCII tape. Once again only the control cards change from the IBM 370 and IBM 360 programs.

4.1.3 UNIVAC 1108 BCD → ASCII Conversion Program

A conversion program was written in UNIVAC assembly to allow for the conversion of BCD record data on the UNIVAC 1108. As with IBM conversion, data is accepted from either cards or 7-track or 9-track magnetic tape. Each character is then converted to the corresponding 7-Bit ASCII character. A REEDA System compatible 9-track ASCII tape is generated as shown in Figure 4-2.

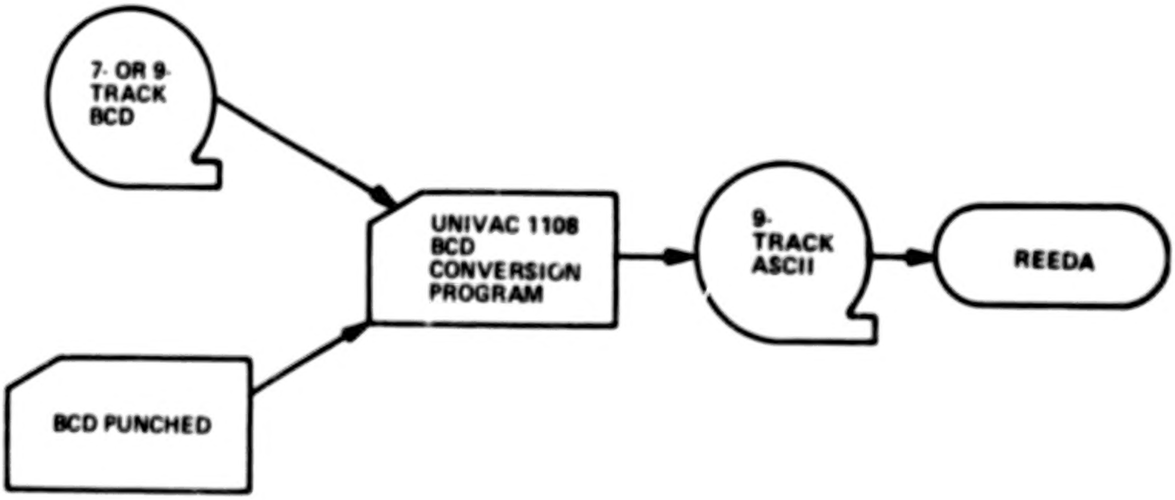


Figure 4-2. UNIVAC 1108 BCD → ASCII Conversion Process

4.1.4 UNIVAC 1108 EBCDIC → ASCII Conversion Program

A conversion program was also written in UNIVAC assembly language to allow for the conversion of EBCDIC recorded data on the UNIVAC 1108. This program is identical to the BCD → ASCII conversion program except all EBCDIC characters are converted to ASCII. As shown in Figure 4-2, input can be either punched cards or magnetic tape with the output being a 9-track ASCII REEDA compatible tape.

4.2 CONVERSION PROGRAMS (SPECIFIC)

All of the generic BCD/EBCDIC conversion programs allow data to be converted from card/tape to REEDA System compatibility in a one-pass operation. However, the conversion programs were developed with a prerequisite that all data records be 80 characters long (i.e., card size). Thus, in the event records being converted (from tape) are not card images, that is, longer or shorter than 80 characters, a pre-processor is needed to reformat the data into 80 character records. This data can then be used as input into the BCD/EBCDIC conversion programs as shown in Figure 4-3.

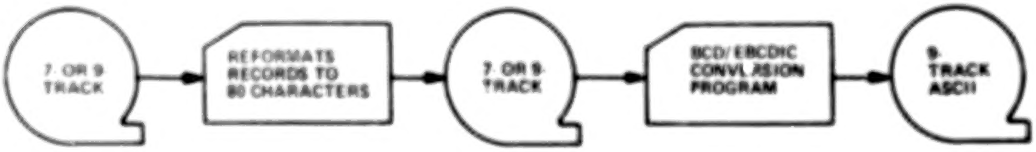


Figure 4-3. Reformatting to 80 Character Records Then Converting BCD/ECBDIC → ASCII

Here the reformatter programs were developed to reformat various data from KSC, JSC, Pt. Mugu, Vandenberg AFB, etc., generated on IBM 7094, IBM 360, CDC 3300, UNIVAC 1108, etc., computers to REEDA compatibility. These programs are discussed in Chapter 5.

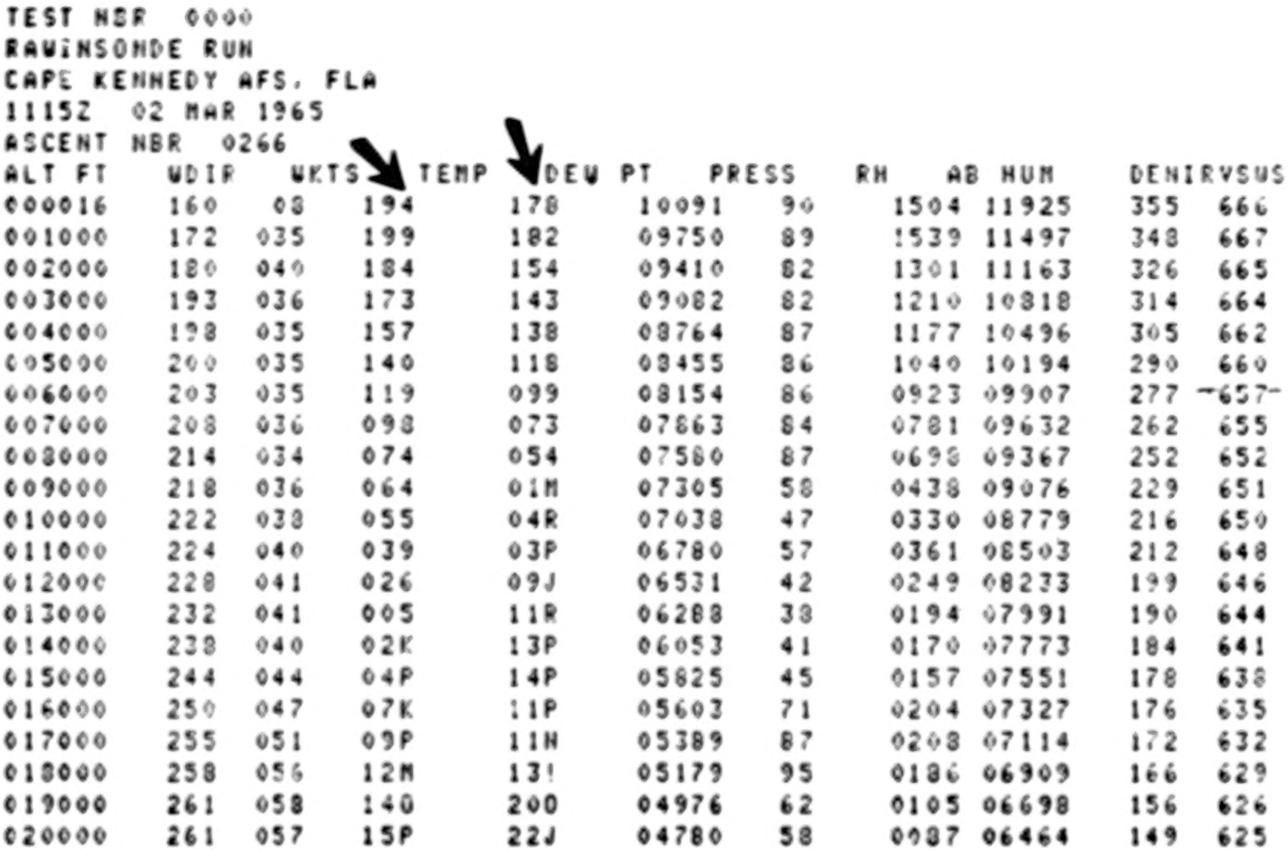
4.2.1 HP EBCDIC → ASCII Conversion Program

A conversion program was written in HP assembly language to translate IBM 9-track, ODD parity, EBCDIC recorded tapes to the compatible HP format. The program is capable of converting all IBM EBCDIC characters into their 7-Bit ASCII equivalent. Each 32 bit IBM integer is translated to a 16 bit HP integer, and 32 bit IBM real numbers are translated into the HP 32 bit real number format. For this program, the user must define the record lengths and blocksize to the conversion program which then translate the tape as it is being processed. Since only one tape drive exists for the current REEDA configuration, it is not possible to convert the entire tape and rewrite it to another tape for subsequent processing, thus the UNIVAC 1108 and IBM 370/360 conversion programs prove more efficient in most instances.

4.2.2 1965 KSC Rawinsonde Data Conversion Program

The conversion of the 1965 KSC rawinsonde data tapes (18 tapes, four recordings a day for twelve months) for REEDA usability was performed. The BCD to ASCII conversion program for the UNIVAC 1108 was utilized to convert from 7-track to the 9-track REEDA format. In initial attempts to process the data, once it was converted to REEDA System format, it was noted (see Table 4-1) that a non-standard method of recording negative numbers was used when recording the original data. That is, a negative number was represented by over punching an (11) in the right most digit of the variable field. Thus, for the numeric values of 0 - 9 together with an (11) punch would give the visual representation as follows:

Table 4-1. Representative Rawinsonde Data



```
TEST NBR  0000
RAWINSONDE RUN
CAPE KENNEDY AFS, FLA
1115Z  02 MAR 1965
ASCENT NBR  0266
ALT FT    WDIR    WKTS    TEMP     DEW PT    PRESS    RH    AB HUM     DENIRVSUS
000016    160      08     194       178      10091     90     1504 11925    355   666
001000    172     035     199       182      09750     89     1539 11497    348   667
002000    180     040     184       154      09410     82     1301 11163    326   665
003000    193     036     173       143      09082     82     1210 10818    314   664
004000    198     035     157       138      08764     87     1177 10496    305   662
005000    200     035     140       118      08455     86     1040 10194    290   660
006000    203     035     119       099      08154     86     0923 09907    277  -657-
007000    208     036     098       073      07863     84     0781 09632    262   655
008000    214     034     074       054      07580     87     0698 09367    252   652
009000    218     036     064       01M      07305     58     0438 09076    229   651
010000    222     038     055       04R      07038     47     0330 08779    216   650
011000    224     040     039       03P      06780     57     0361 08503    212   648
012000    228     041     026       09J      06531     42     0249 08233    199   646
013000    232     041     005       11R      06288     38     0194 07991    190   644
014000    238     040     02K       13P      06053     41     0170 07773    184   641
015000    244     044     04P       14P      05825     45     0157 07551    178   638
016000    250     047     07K       11P      05603     71     0204 07327    176   635
017000    255     051     09P       11N      05389     87     0208 07114    172   632
018000    258     056     12M       13!      05179     95     0186 06909    166   629
019000    261     058     140       200      04976     62     0105 06698    156   626
020000    261     057     15P       22J      04780     58     0087 06464    149   625
```

81!	implies	-810
81J		-811
81K		-812
81L		-813
81M		-814
81M		-815
81O		-816
81P		-817
81Q		-818
81R		-819

Thus a program was developed to process the data tapes to restore the number back to usable numeric notation. These tapes were then used as input to the REEDA System to build a "single tape" data base containing all pertinent information from the existing 18 tapes. The "single tape" data base allows the user easier/faster access to any/all data which he desires to process, thus eliminating the need to keep a library of 18 tapes and the processing of data which is not desired (See Figure 4-4).

The "single tape" data base was created by processing each of the 18 tapes and eliminating all data above 20,000 feet in altitude for the standard and mandatory levels, thus eliminating a large portion of the data. This data base was updated after each tape was processed with an EOF (end-of-file) mark inserted at the end of each month. Thus a user can easily access any month from the "single tape" data base by skipping the appropriate number of files. It must be pointed out that the initial idea of a "single tape" data base actually turned out to be two tapes containing all selected information from the original 18 tapes. The first tape contains JAN - JUN 1965, while the second tape contains JUL - DEC 1965.

It should also be noted that each of the 18 tapes being processed require about 2 hours to process on the REEDA System. Thus the initial creation of the "single tape" data

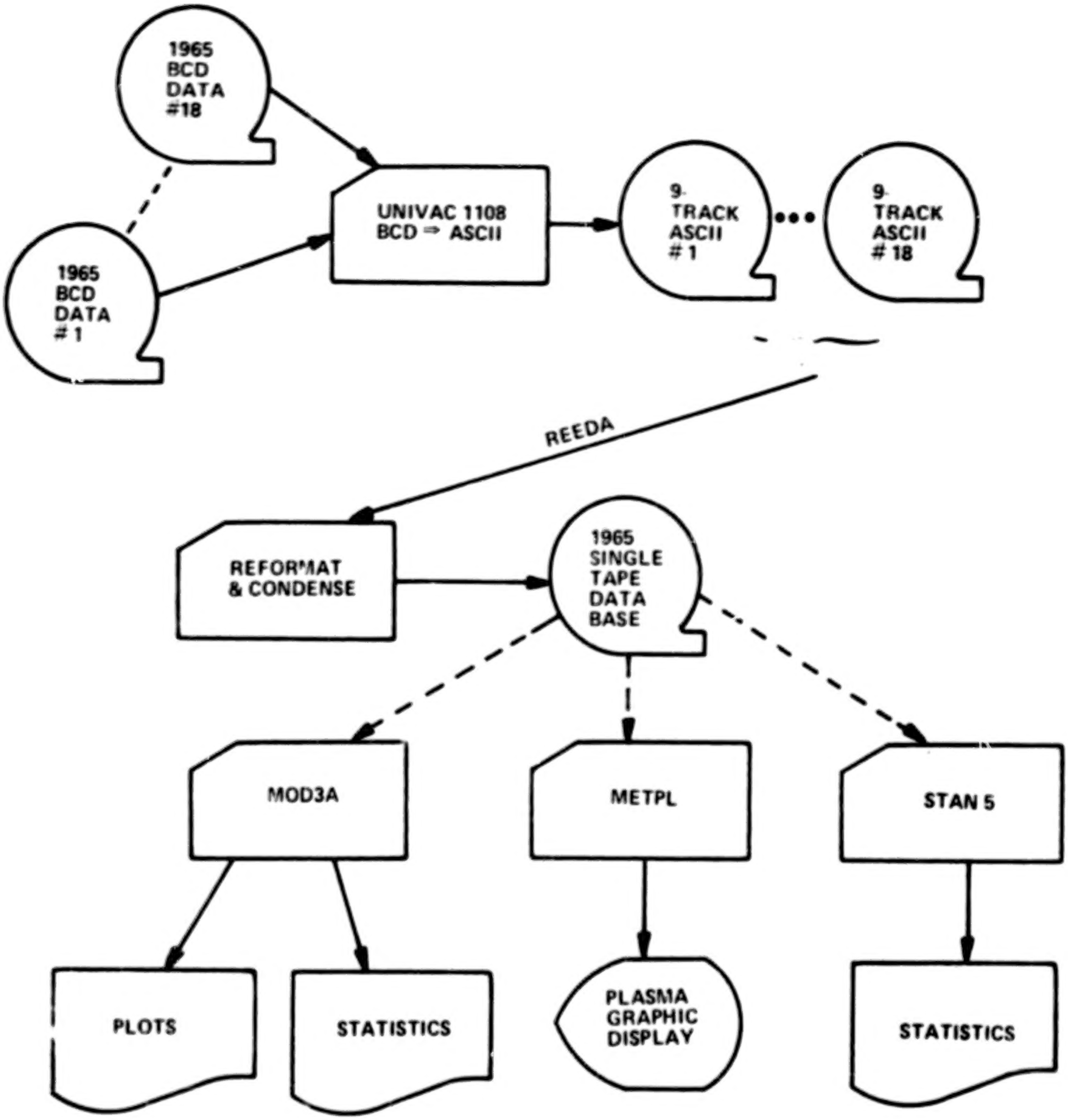


Figure 4-4. 1965 Rawinsonde Data Base Generation and Processing

base was quite time consuming, yet since it occurred only once, while the end product (i.e., "single tape" data base) is being utilized quite frequently. The overall effect of the "single tape" data base for the REEDA System is to allow the user to feasibly process all or as much of the 1965 KSC Rawinsonde data as efficiently and as fast as possible and to eliminate the handling of unneeded and unusable data.

Various programs are now being developed to extensively process the 1965 data, such as MOD3B, METPL, and STAN5 which are all documented in Chapter 5.

4.2.3 1974 Vandenberg Rawinsonde Data Conversion Program

The 1974 Vandenberg AFB Rawinsonde data tape contained two soundings per day (0000Z and 1200Z) for the entire year. The initial task was to convert the data into a usable format for the REEDA System. The original tape was generated on an IBM 360/44 and had variable length/variable block size records with half word alignment. A preprocessor was written in FORTRAN to restructure the data into fixed length records to be used as input into the IBM 370 ECBDIC → ASCII conversion program. It was discovered that two types of data records existed on the tape, 1) PIBAL and 2) Rawinsonde. However, neither data record contained all the information that was required to process the data using the REEDA diffusion model program MOD3A. The PIBAL record contained pressure, altitude, wind speed, wind direction but not temperature. The Rawinsonde records contained pressure, altitude, temperature, dew point, but not wind speed or wind direction. Subsequently, code was generated to merge the two records by means of various interpolations and calculations. The program computed the best possible values at the nearest altitude, pressure, and temperature. Figure 4-5 gives a brief flow of operations for processing the 1974 Vandenberg AFB data.

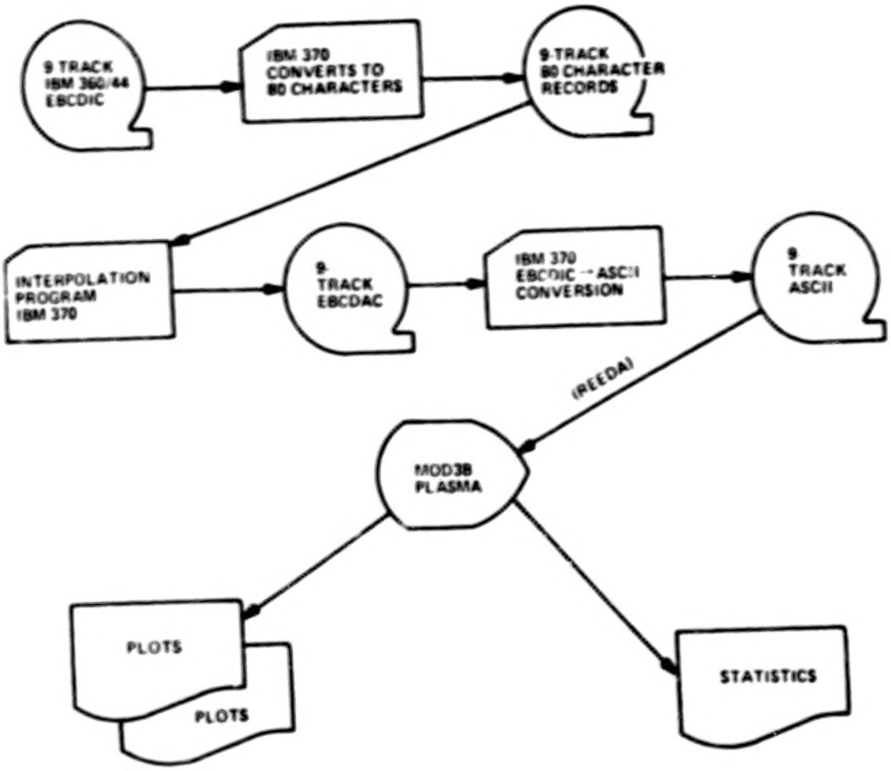


Figure 4-5. 1974 Vandenberg AFB Rawinsonde Data Converting and Processing

Once the data was converted to a usable format for the REEDA System, the program MOD3B was used to process the data. It should be noted that the two soundings per day for the entire year of 1974 were contained on the initial Vandenberg AFB tape. Some 48 cases roughly a week apart at 1200Z hours were processed. The program MOD3B is identical to the program MOD3A except it operates on the Plasmascope, which is interfaced into the REEDA System. It allows faster processing due to the use of "Touch Panel" program options. The output of MOD3B was 48 center line concentration plots and 48 isopleth plots (see Figure 4-6 and 4-7 respectively). This same data will also be processed on the REEDA System utilizing the new version of the program MOD3A.

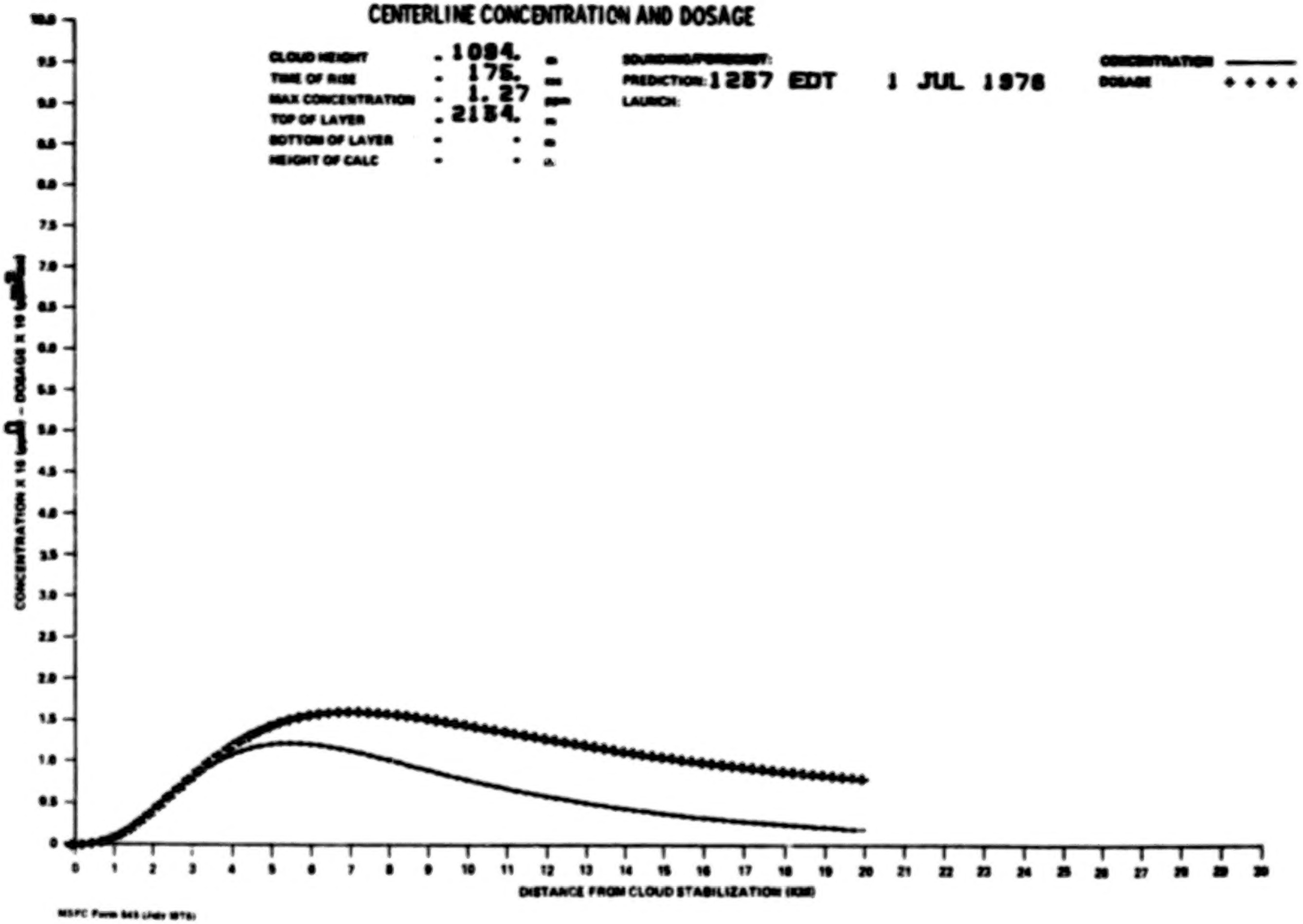


Figure 4-6. 1974 Vandenberg AFB Centerline Concentration and Dosage Plot

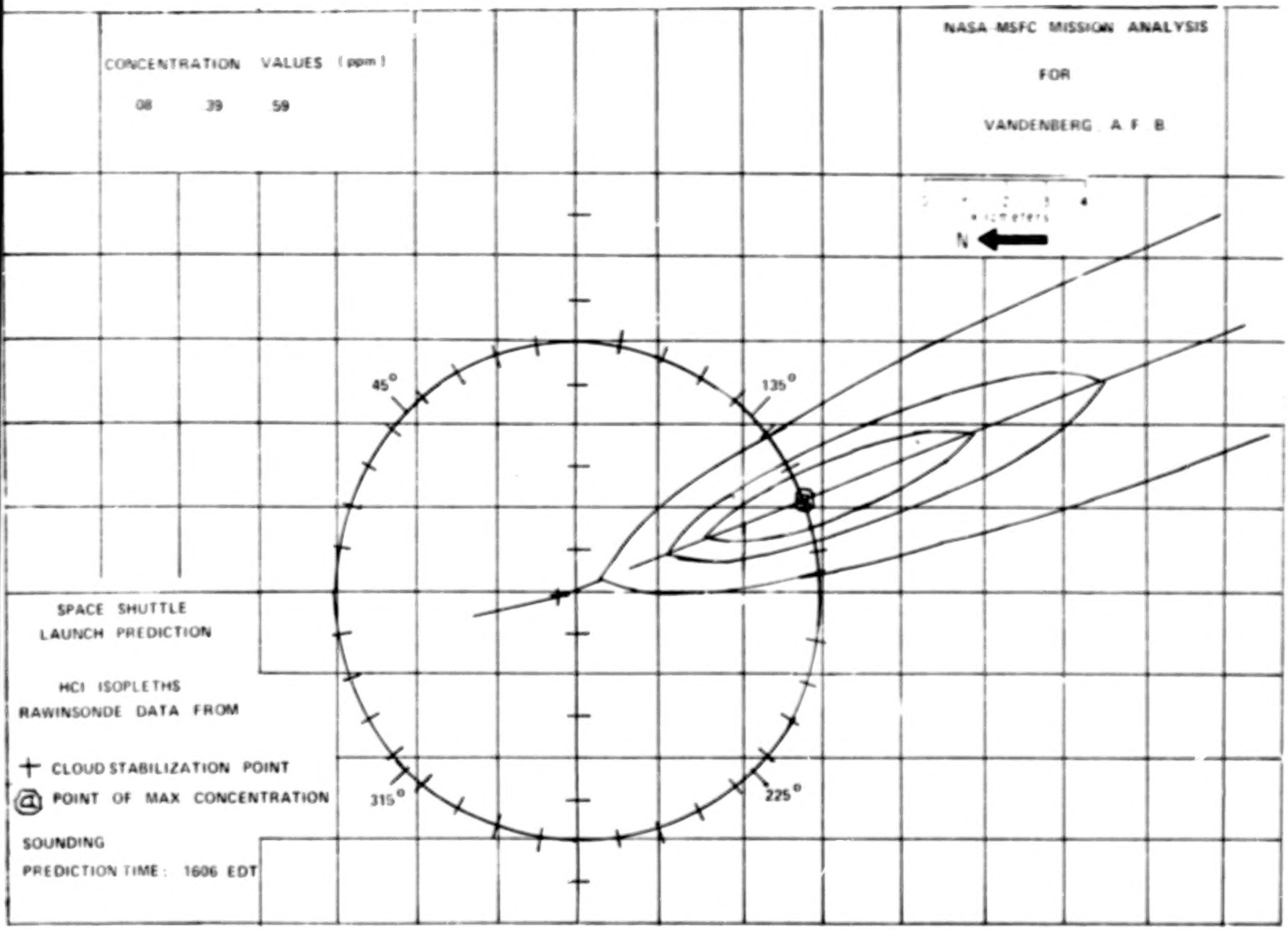


Figure 4-7. Vandenberg AFB Isopleth Contour Plot

4.2.4 1964 - 1970 Jimsphere Data Conversion Program

A conversion program was written in FORTRAN to convert three meteorological data tapes, (1964 - 1966 KSC, 1967 - 1970 KSC, 1965 - 1970 Point Mugu) containing Jimsphere wind data, to the REEDA System compatability. The tapes were initially created on an IBM 7094 with 36 bit word and data written in both fixed point and floating point binary. Each record contained 298 words with 20 such records per file. Additionally, each tape contained from 266 to 294 files. The decision was made to only extract and convert the needed data to eliminate the cumbersome task of processing over and around data not needed for calculations. Only the time, date, altitude, wind direction, and scalar wind speed was deciphered from the original data. It should be noted that the wind speed and wind direction were recorded at equal intervals in altitude from 25 meters to 20,000 meters. Thus, some 800 data points for both wind speed and wind direction were recorded for every Jimsphere profile.

The conversion program was written for the UNIVAC 1108 utilizing both ENTRAN and ENCODE features to convert the tapes into a format usable by the previously built EBCDIC and ASCII program. This encompassed converting from 36-bit to 16-bit HP word size, restructuring data into 80 character fixed length records, eliminating unwanted data, and then converting to ASCII format as shown in Figure 4-8.

As can be seen, a program to process the Jimsphere wind data on the REEDA System was created called JIMPL which will be discussed in Chapter 5.

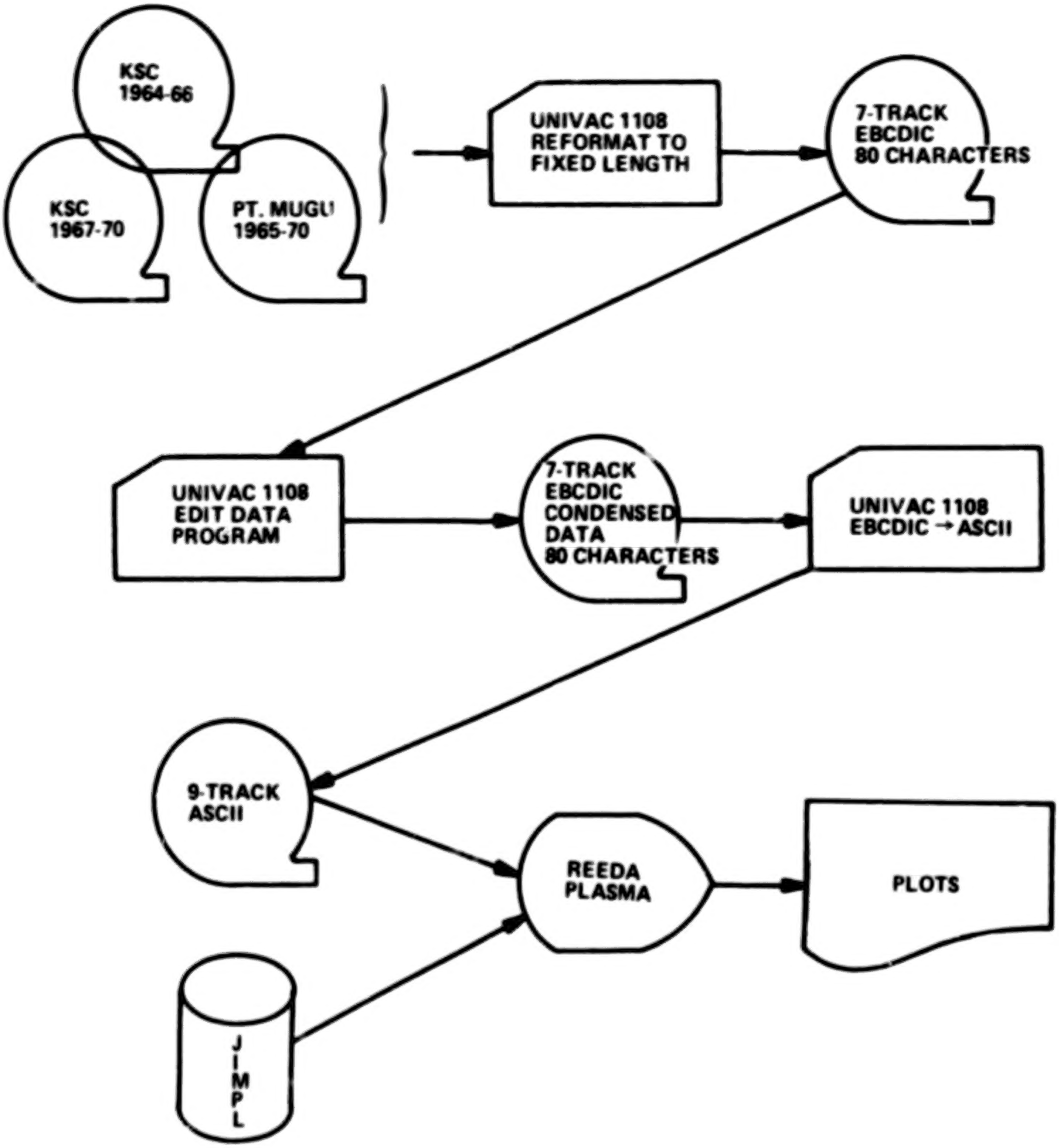


Figure 4-8. Jimsphere Data Conversion and Processing

4.3 CONCLUSIONS AND RECOMMENDATIONS

These conclusions and recommendations are based on the conversion programs documented in Sections 4.1 and 4.2 respectively. In reviewing the generic conversion programs that were developed, it seems that they have proven quite satisfactory in providing a mean for converting non-standard REEDA formatted data into a usable form. Data generated on almost any other computer configuration, either 7-track or 9-track, either BCD or EBCDIC, either fixed or variable length records, can be made compatible to the REEDA System via one or a combination of two or more of the conversion programs that have been developed. However, it is still probable that data will be acquired that cannot be directly converted by using just the available conversion programs to date. Consequently, additional conversion programs undoubtedly will be developed as required.

It is also recommended that in most instances, tape reformatting and tape converting be conducted on a large scale computer configuration where multiple tape drivers and faster operating speeds are available.

5. INTERACTIVE REEDA PROGRAMS

In this section all of the applicable interactive REEDA programs are discussed. A brief description, along with current and future applications of each program is given. The following is a list of all the current REEDA programs which are discussed in Sections 5-1 thru 5-7.

- MOD3A
- MCD3B }
- METPL } REED Program*
- STAN5 }
- MIXH
- JWSPL
- JWDPL
- JIMPS
- SKEW T (Version I & II)

5.1 MOD3A

The HP 9820 breadboard version of the REED Description, Model 3, previously used to monitor launches (46-49) has been rewritten, liberally commented, and made research operational on the REEDA System as an interactive program to test human factors and provide a real-time research capability for surface air quality predictions. The program asks questions of the user and accepts answers in English words and phrases. Using an X-Y plotter, it draws concentration and dosage versus distance plots as well as isopleth contour plots. The equations used for the cloud rise and diffusion are in an extremely simplified form and are being expanded to give a more accurate representation of the cloud mechanics and the diffusion process. This version does not permit diffusion calculations aloft, does not allow for options like surface absorption, rain scavenging, or Al_2O_3 deposition.

*These have been merged into the NASA/MSFC REED Diffusion Model Program Version I.

The distinct advantages of having the diffusion model operational on the REEDA System are two-fold. First, because the system is dedicated, the program can be ru in almost real-time, thereby allowing last minute analysis and decisions to be made. Secondly, because of the interactive nature of the program, it is not necessary to have a trained computer person run it. Any scientific person knowledgable in diffusion theory can, with a brief orientation, successfully operate the program. Knowledge of diffusion theory is required because SIGAR and the top of the transport layer determination calculations have not yet been automated.

5.2 MOD3B

A version of MOD3A, called MOD3B, has been written for the REEDA System to use the Plasmascope installed on the system. Because of the Touch Panel feature on the scope, it is easier for the user to answer the yes/no type questions asked him by the program. He need only touch the YES or NO area on the screen instead of typing in the answer. Further man-machine interface improvements using the Plasmascope are planned for MOD3B to make the program, both input and output, as simple and informative as possible.

5.3 JIMPL

A program, written in FORTRAN, to process the Jimsphere wind data was created on the REEDA System. This program produces both scalar wind speed and wind direction plots. An example of each is given in Figure 5-1 and Figure 5-2 respectively. This program requires as input the following data on altitudes ranging from 25 to 20,000 meters.

- Time
- Date
- Altitude
- Wind Direction
- Scalar Wind Speed

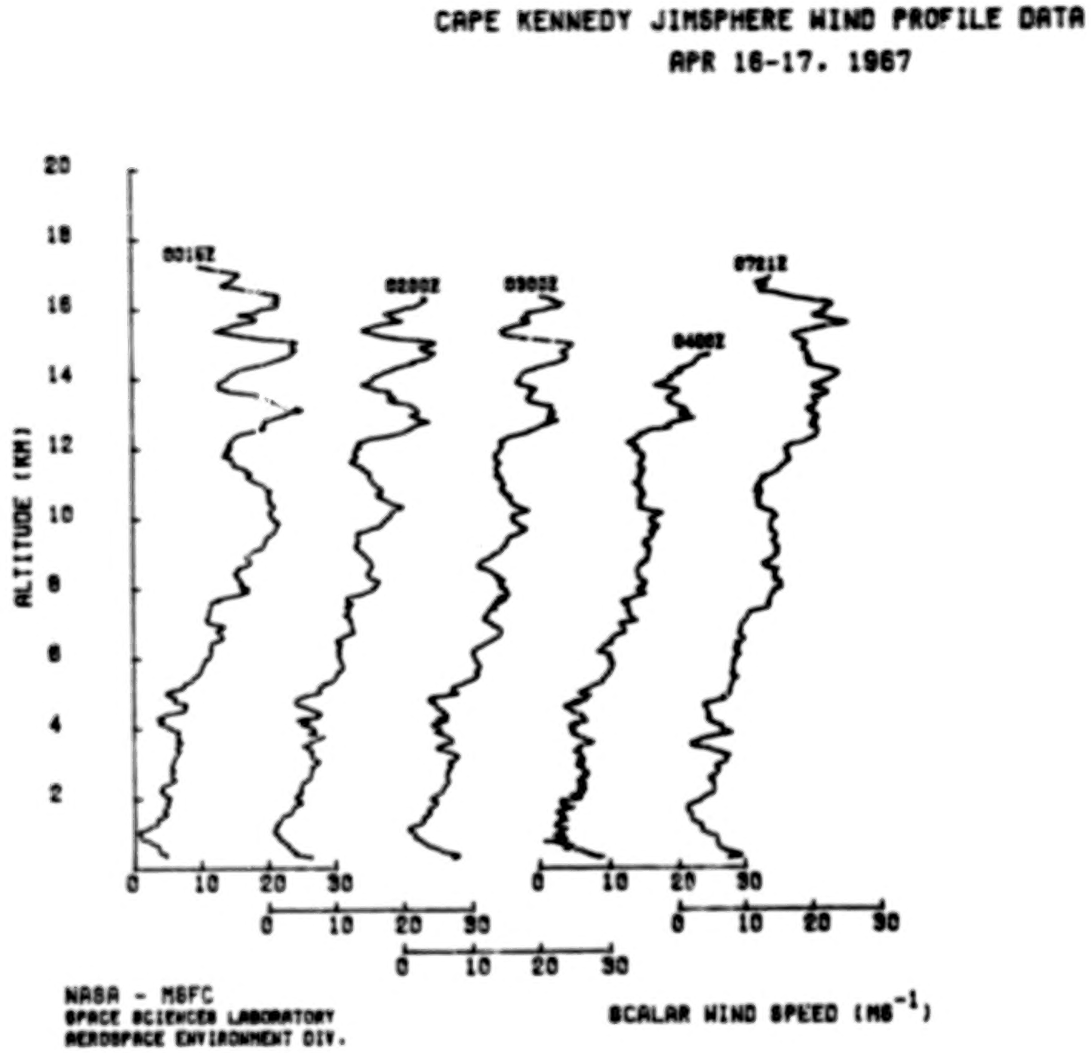


Figure 5-1. Example Jimsphere Scalar Wind Speed Plot

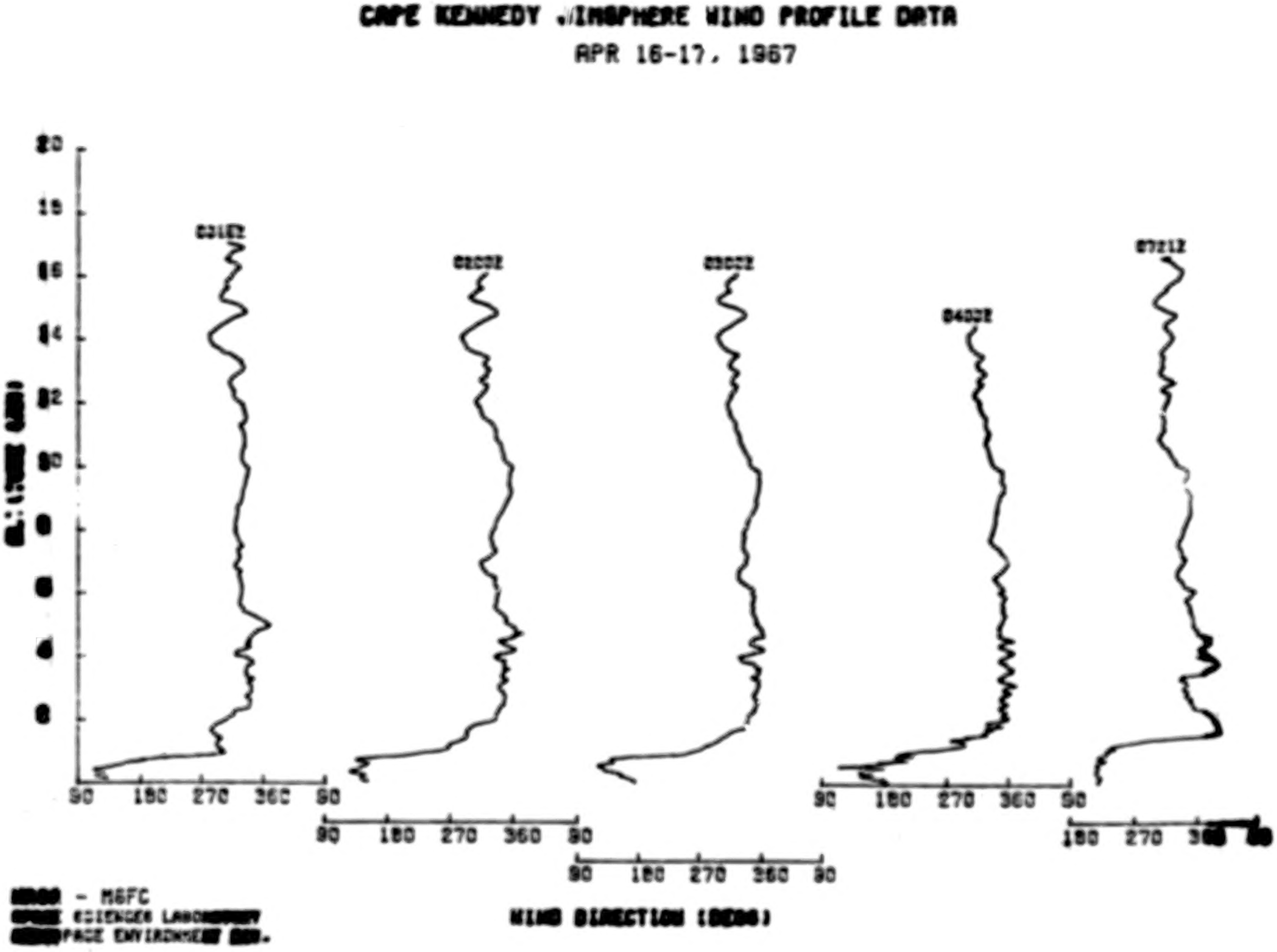


Figure 5-2. Example Jimsphere Wind Direction Plot

This program was used to initially process the KSC and Point Mugu Jimsphere wind data to determine temporal variation in the atmospheric kinematics to support climatic diffusion assessments. Over 150 plots were generated. Several modifications were found desirable such as the inclusion of a filter to eliminate bad data or noise, and provide the ability for the user to select only those specific profiles of interest. The ultimate desire was to be able to see a plotted profile without having to actually plot it. This would allow the user to process only data of interest and allows the creation of final Jimsphere profile plots without having to process the data several times.

Consequently, programs to process Jimsphere wind speed/direction (JWSPL, JWDPL and JIMPS) were developed for the REEDA Plasmascope which allows for "touch panel" control. Research or production options are available to allow the user to process all, or portions, of the data with graphic plots on the Plasmascope or hard copy plots. Program JIMPS allows the user to visually display a complete Jimsphere wind speed direction plot on the Plasmascope, thus allowing the user to quickly scan and edit the data before making a hard copy plot. This feature ensures the ability to only generate useful hard copy plots.

The programs JWSPL and JWDPL process the Jimsphere Wind Speed/Direction data respectively. Each allows the user to easily process Jimsphere data quickly through "touch panel" questions and answers, thus eliminating the possibility of erroneous keyboard input. An example scenario from the program JWSPL is given in Figure 5-3. Hopefully, it can be seen from the scenario that by using the Plasmascope "touch panel" control the possibility of making input errors (keyboard) are reduced. A noncomputer oriented user can easily be taught to use such a program within minutes.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONCLUDED)

Section Page

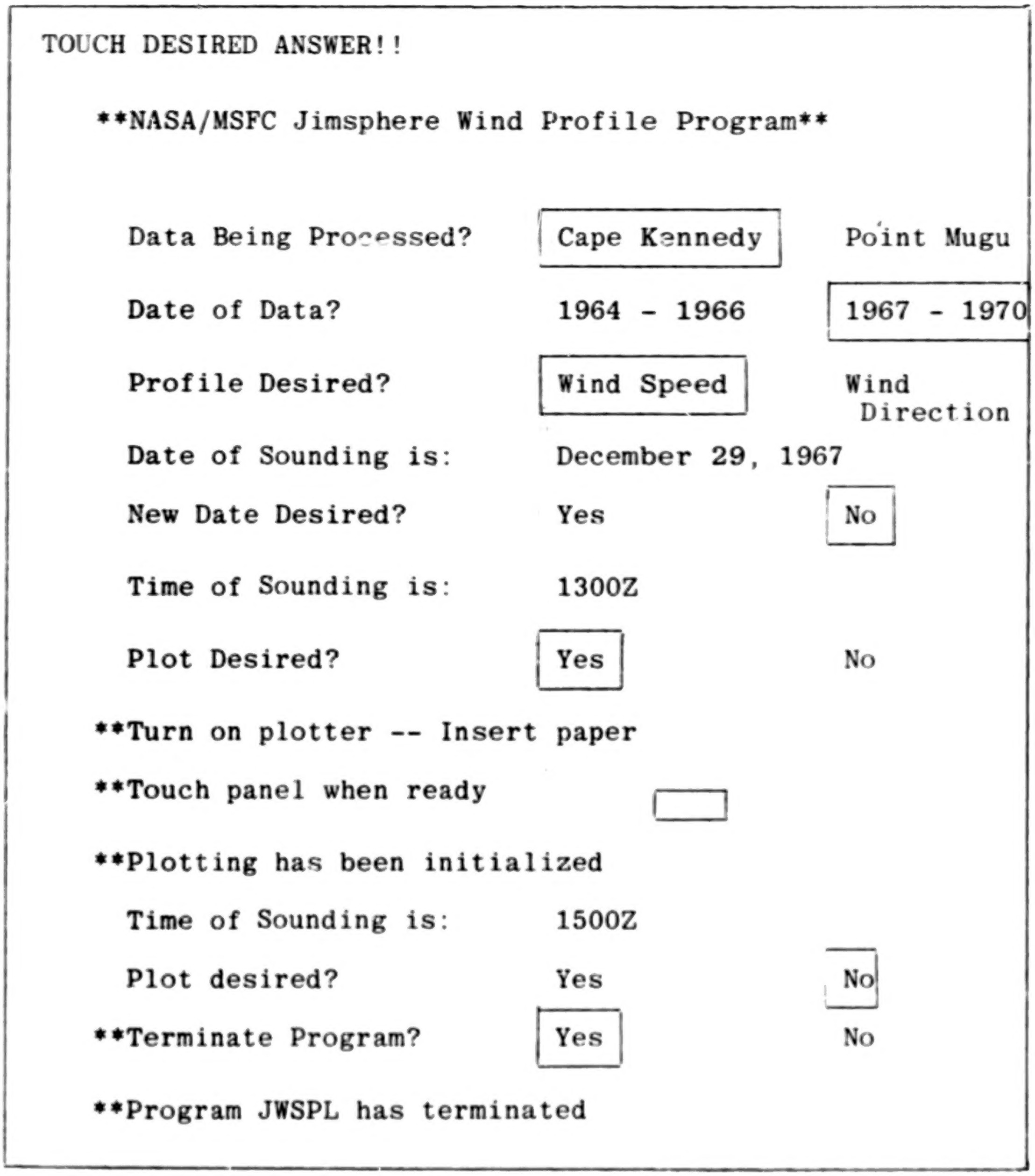
LIST OF ILLUSTRATIONS

Figure — Page

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

Table Page



```
TOUCH DESIRED ANSWER!!

   **NASA/MSFC Jimsphere Wind Profile Program**


     Data Being Processed?       [✓ Cape Kennedy]    Point Mugu

     Date of Data?               1964 - 1966     [✓ 1967 - 1970]

     Profile Desired?            [✓ Wind Speed]      Wind
                                                      Direction
     Date of Sounding is:        December 29, 1967

     New Date Desired?           Yes               [✓ No]

     Time of Sounding is:        1300Z

     Plot Desired?               [✓ Yes]             No

   **Turn on plotter -- Insert paper

   **Touch panel when ready            □

   **Plotting has been initialized

     Time of Sounding is:        1500Z

     Plot desired?               Yes               [✓ No]

   **Terminate Program?          [✓ Yes]             No

   **Program JWSPL has terminated
```

Figure 5-3. Example of Operating JWSPL Plasmascope Program

The overall results of the Plasmascope Jimsphere programs have been quite successful and time saving. Over 700 finalized Wind Speed and Wind Direction profiles have been generated from the KSC and Point Mugu data.

5.4 METPL

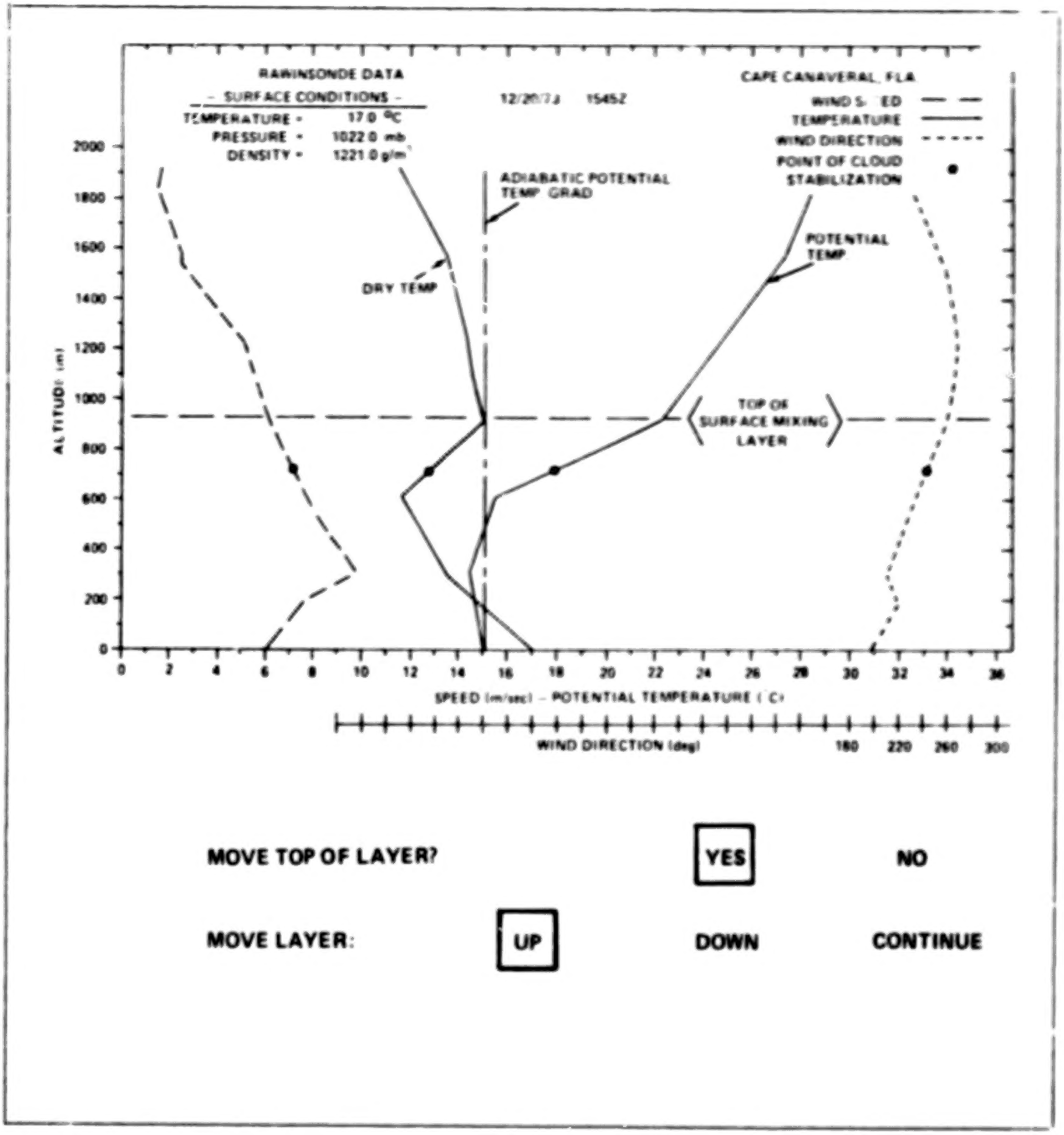
METPL is currently a stand alone research program generated to allow visual display of Wind Speed, Wind Direction, Dry Temperature, Potential Temperature, and Cloud Stabilization Height as one profile upon the Plasmascope. This program should be interfaced to MOD3B. As an example of the meteorological profile as it appears on the Plasmascope is shown in Figure 5-4. Various questions will appear at the bottom of the Plasmascope to direct the user as to the moving up or down of the top of the surface mixing layer to the desired height as well as giving the option for a hard copy plot of the generated profile. The meteorological profile is normally obtained from a Rawinsonde of the atmosphere. To obtain the entropy profile required for these soundings, the temperature and pressure are translated into the potential temperature in accordance with the following equation:

$$\theta = T\left(\frac{1000}{p}\right)^{0.288}$$

where the concept of a potential temperature (θ) is introduced to reference the temperature to a specific pressure (1000 mb).

5.5 STAN5

Program STAN5 is a research stand alone program written in FORTRAN and operates on the REEDA System. This program should be interfaced to the MOD3B program. STAN5 calculates the standard deviation of the horizontal wind azimuth angle, SIGAR. Input data are the temperatures, pressures, and altitudes of the first three data levels of KSC Rawinsonde soundings. The levels are the first and second standard



RAWINSONDE DATA

- SURFACE CONDITIONS -

TEMPERATURE = 17.0 °C

PRESSURE = 1022.0 mb

DENSITY = 1221.0 g/m

12/20/73 1545Z

CAPE CANAVERAL, FLA

MOVE TOP OF LAYER? [✓ YES] NO

MOVE LAYER: [✓ UP] DOWN CONTINUE

Figure 5-4. Example of METPL Plasmascope Program

altitude levels (16 and 1,000 feet) and the first mandatory pressure level (1,000 mb). The roughness length along the air trajectory with the surface transport layer is also an input variable. The background for the calculation is described in Section 2.

The output o. STAN5 includes the time and date of the Rawinsonde sou ding, the input data, calculated non-dimensional parameters, the gradient of potential temperature, and SIGAR.

5.6 MIXH PROGRAM

Program MIXH is a stand alone research program which operates on the REEDA system. MIXH selects a surface transport layer height based on criteria described in Chapter 2. The input data is a Rawinsonde sounding. MIXH calculates virtual temperature at each level and tests the data according to the prescribed criteria for virtual temperature gradient corresponding to the base of a stable layer and the top of a stable layer. The layers must have a thickness of at least 100 meters. The base of the stable layer nearest to the ground is offered as the height of the surface transport layer. If a stable layer is ground based, then the top of the stable layer is selected as the transport layer height. If no stable layers are found between the surface and 3000 meters, then the transport layer height is set equal to 3000 meters. The output of MIXH is the mixing height of the surface transport layer. If the theory is upheld by extensive test, it should be interfaced to the MOD3B program.

5.7 SKEW T

The existing SKEW T REEDA System program, originally written by Dr. J. B. Stephens ·f MSFC, was modified to enhance its capabilities in processing sounding data. It can current'y accept sounding data ırom both magnetic tape or disc and in a variety of user specified formats. The SKEW T program generates logarithmic plots for both dew point and temperature as a function of altitude as shown in Figure 5-5.

The SKEW T program was used to process some 23 cases of Battelle Thiokol data for 1974. One additional modification was made to the SKEW T program to allow for processing the Battelle data, which was the calculation of dew point from a given relative humidity and temperature.

5.8 CONCLUSIONS AND RECOMMENDATIONS

These conclusions and recommendations are drawn from the discussion of the interactive REEDA software described in Section 5-1 thru 5-7. It should be evident that a variety of sophisticated interactive REEDA software has been generate. and utilized during this contractual period. A vast amount of data from various sources have been processed, analyzed, plotted, etc., by the different REEDA programs. The REEDA software has proven effective, efficient, and invaluable in providing both fast/accurate results in both statistical and graphical form. The current REEDA software, especially the Plasmascope programs provide a means for even a non-computer oriented user to operate and get results with very little effort. The Plasmascope "Touch Panel" capability provides n t only for faster user response (i.e., touch-vs-keyboard) but proves superior to the CRT program due to the fact it virtually eliminates or safeguards the user from entering an erroneous ·alue/answer. In addition, due to the 512 by 512 raster dot resolution provided by the Plasmascope virtually

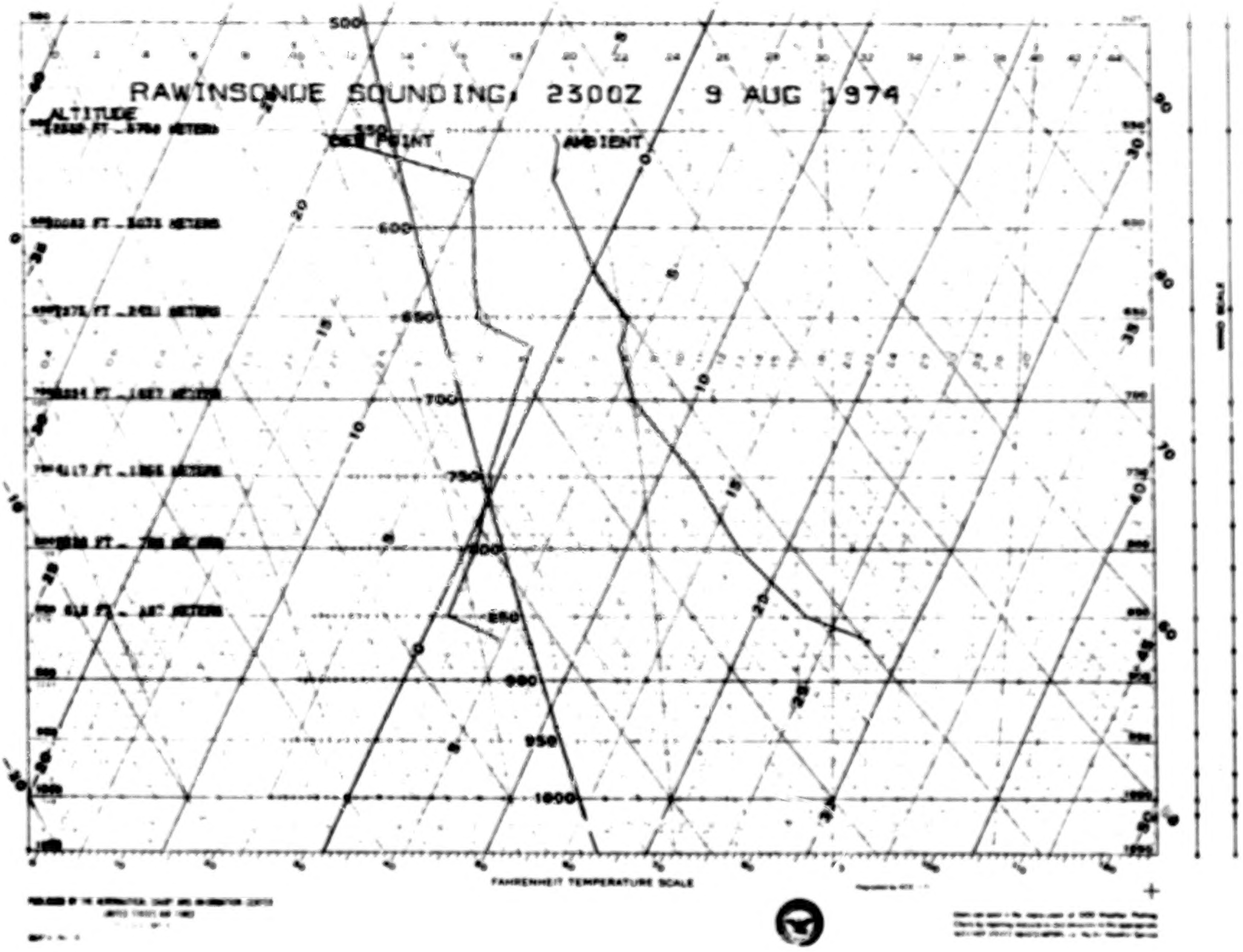


Figure 5-5. Example of SKEW T Plot

unlimited visual graphic display can be generated.

It is recommended that all existing stand alone REEDA software be extensively tested and validated to its fullest extent, with state-of-the-art Plasmascope technology being incorporated whenever and wherever feasible. Additional REEDA software should also be developed, utilizing the REEDA Plasmascope technology, to provide even more capabilities in processing both present and future sources of data.

6. NUMERICAL CLOUD RISE MODEL

Under contract with the Army Missile Command High Energy Laser Programs Office, SAI has recently developed a digital computer program (PUFF) representing a first-order mathematical model for describing the behavior of clouds produced by short-duration high temperature exhausts. In order to more clearly identify and understand the important features associated with the problem of cloud behavior, the cloud history was divided into three phases as depicted in Figure 6-1 and as tabulated in Table 6-1. As indicated in the figure and table, the cloud's history from the time of its initial formation until it reaches equilibrium altitude is contained within Phases I and II. PUFF was primarily designed to handle the problem of cloud behavior during these two phases.

The basic model upon which PUFF is based is the result of a study of relevant literature, both theoretical and experimental. In essence, the cloud is treated as an open thermodynamic system within which all properties are assumed to be uniform. The cloud shape is represented by a sphere and cylinder combination as shown i Figure 6-2. The cloud behavior is predicted by the simultaneous numerical solution of the

- Conservation equations for
  1) Mass
  2) Momentum (3 components)
  3) Energy
- Equation of state
- Volume and center of mass relations for cylinder and sphere combinations.

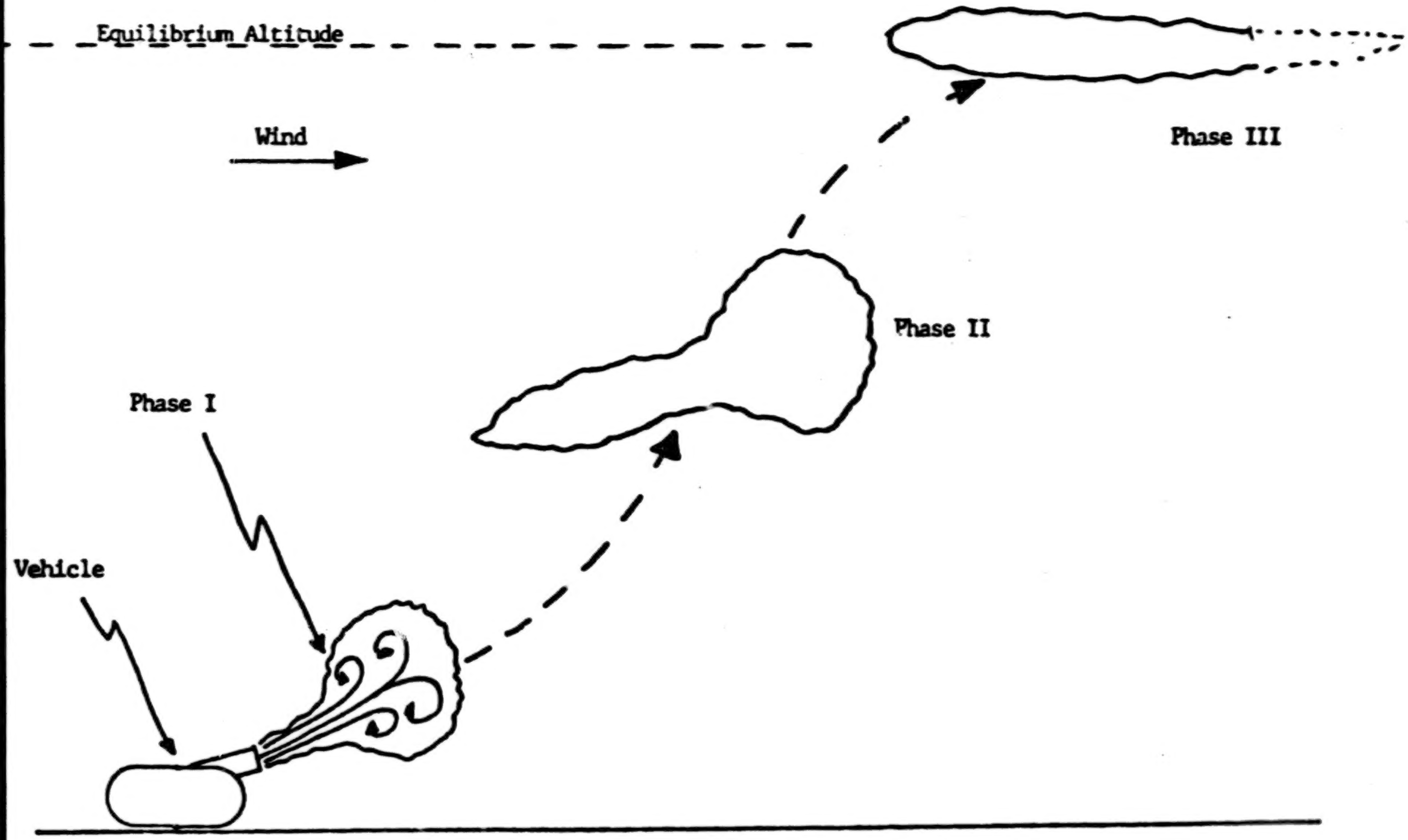
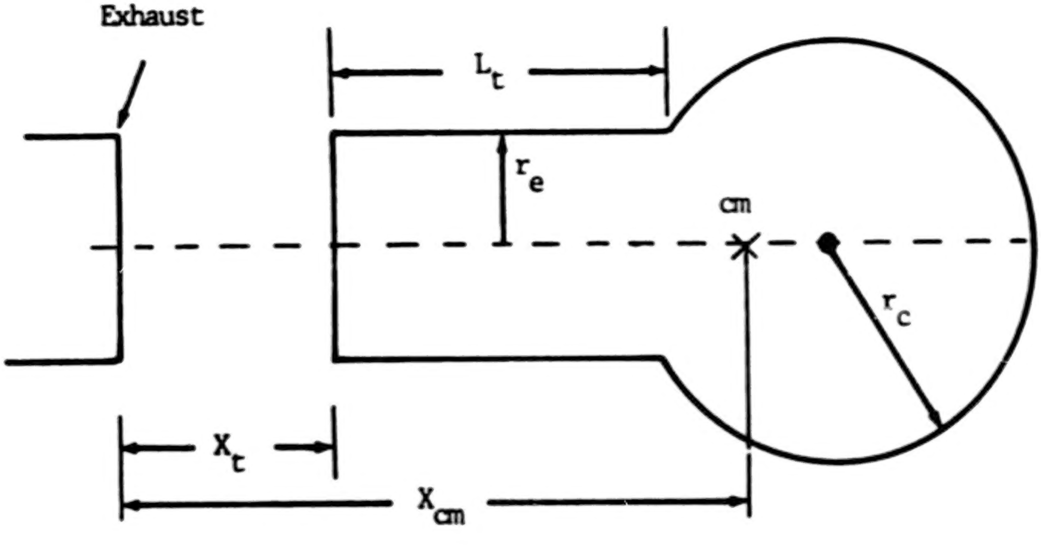


Figure 6-1. General Representation of the PUFF Program

Table 6-1. Phases of the PUFF Program

Phase		Cloud Behavior	Dominant Effect	Other Effects Present
I	(Thermodynamic Phase)	Vortex ring with tail formed near vehicle	Exhaust momentum flux	Buoyancy, Drag, Diffusion
II		"Tadpole" shaped cloud rises through atmosphere	Buoyancy	Drag, Diffusion
III	(Kinematic Phase)	Cloud reaches equilibrium altitude and spreads out	Diffusion	Drag



L_t - length of tail

r_c - radius of sphere

r_e - radius of exhaust

X_t - distance from exhaust to end of tail

X_{cm} - distance from exhaust to center of mass

Figure 6-2. Cloud Shape

The resulting solution yields

- Position (x,y,z)
- Velocity (U_{c1}, U_{c2}, U_{c3})
- Temperature
- Density
- Shape*

of the cloud as a function of time

6.1 MODIFICATIONS TO PUFF

Some modifications to PUFF are necessary to allow it to be applied to the situation shown in Figure 6-3. Some of the changes are general, relating to both the duct cloud and the ground cloud, while other changes deal with the ground cloud only.

6.1.1 General Changes

General changes to the program include (1) the introduction of atmospheric density and temperature profiles, (2) the calculation of energy released by chemical reaction, (3) the calculation of thermal radiation emitted by the exhaust products and (4) the calculation of the behavior of liquid droplets and solid particulates suspended in the exhaust gases.

The use of atmospheric density and temperature profiles would be based on atmospheric data obtained from soundings. Soundings are taken at regular time intervals before each firing and twice a day normally.

The calculations of the energy released by chemical reaction would involve maintaining an inventory of the chemical species present in the cloud and computing the rate and total amount of each significant reaction associated with the production or consumption of each species. The techniques used in the afterburning analysis and for maintaining an inventory of chemical species have been discussed in Section 3.2.

*In terms of length of cylindrical tail and radius of spherical body.

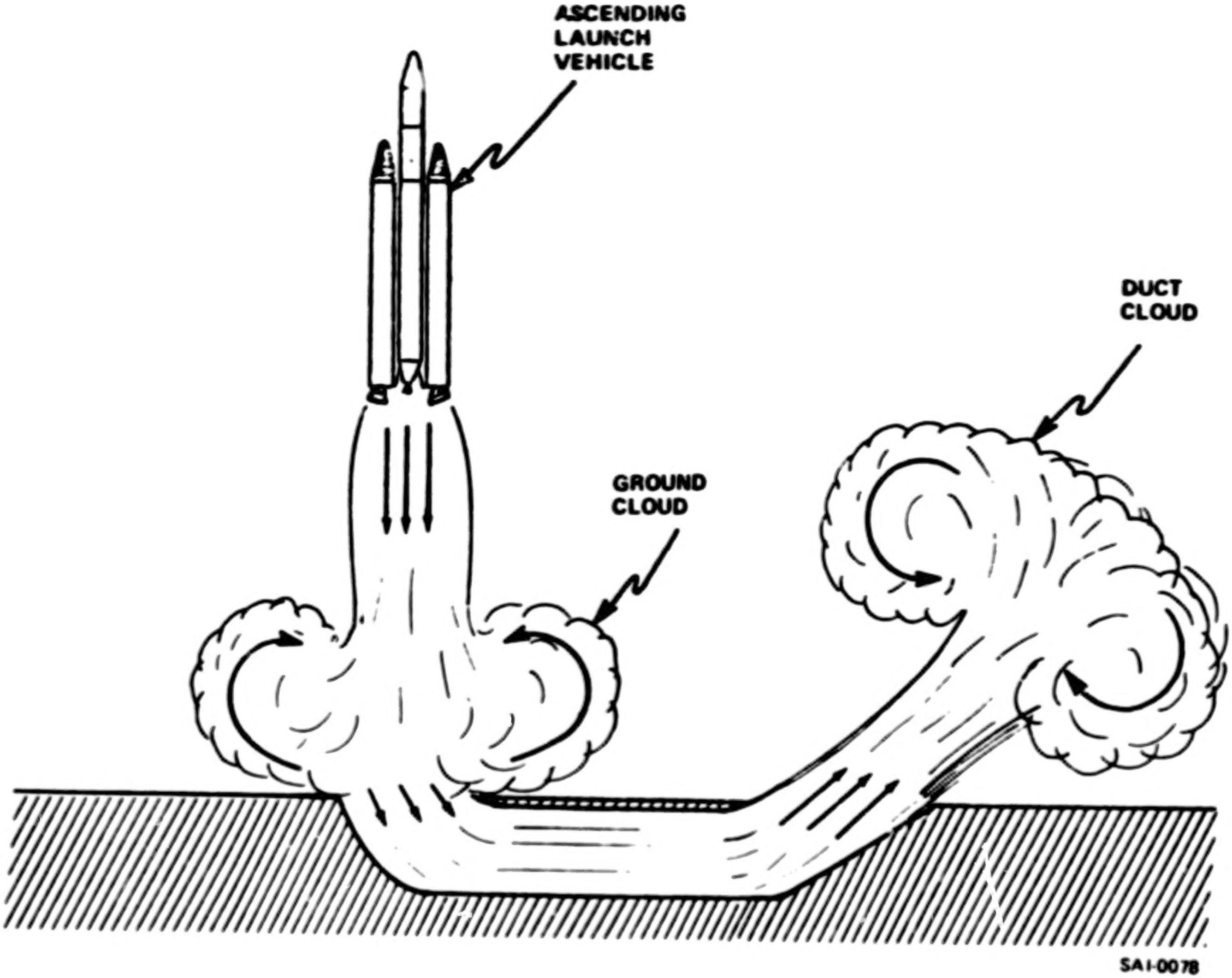


Figure 6-3. Exhaust Cloud Configurations

The spectral characteristics of the thermal radiation emitted by the exhaust products (gaseous, liquid droplets, and solid particulates) are quite complex and calculation of such characteristics is not a simple task. The total radiation emitted by the various constituents, however, can be calculated by standard engineering techniques and should prove adequate for the type of model under consideration.

The behavior of liquid droplets and/or solid particulates within the cloud depends upon the size of the droplets/particles and the velocity of the flow. Methods for predicting such two-phase flow phenomena have been presented in Subsections 3.1 and 3.3.

6.1.2 Ground Cloud Changes

In addition to the four general changes noted in the preceding subsection, there are certain modifications which relate specifically to the ground cloud alone. PUFF was not originally designed for the case where the rocket exhaust impinges on a solid surface. The program can be easily modified such that in the presence of a solid surface, a surface force is introduced into the momentum equations in such a way that the ground cloud center-of-mass cannot pass through the surface.

Although the ground cloud center-of-mass does not penetrate the surface, allowance must be made for mass, momentum, energy, and species to escape from the ground cloud through the flame trench entrance and ultimately into the duct cloud. All such losses to the ground cloud would be added to the duct cloud to satisfy basic conservation principles. Calculations of the magnitude of the losses would depend upon the height of the rocket engines above the flame trench entrance. As the launch vehicle ascends, the amount of exhaust gases passing through flame trench entrance decreases. This decrease results from the vertical rocket exhaust plume cross section (at ground level) increasing with time while the velocity within the plume (as ground level) is decreasing.

In the original version of PUFF, the origin of the coordinate system was the center of the jet exit plane. For the case of the ground cloud, this origin would move upward as the launch vehicle ascends. Because it is desirable to have an origin which is stationary with respect to the ground level, PUFF must be modified such that the origin remains fixed at a point corresponding to the center of the rocket exhaust exit plane prior to liftoff. The rocket exhaust exit plane after liftoff will be programmed to move upward in accordance with the known trajectory of the launch vehicle.

Another modification to the program would involve the manner in which the buoyant force is calculated for the ground cloud. Currently in PUFF the buoyant force depends on the difference between the mean cloud density and the atmospheric density at ground level. The atmospheric density surrounding the tail of th ground cloud varies with altitude and thus the buoyant force should involve an integral of the density difference over the altitude interval from ground level to the end (top) of the ground cloud tail.

6.2 CURRENT STATUS OF PUFF PROGRAM

The PUFF Program was converted from the IBM 370 to execute on the REEDA System. Various software incompatibilities had to be resolved, such as:

- Label common not supported
- Multiple entry points not supported
- Initialization of common variables in data statements not supported
- Block data not supported
- Namelist read not supported

These were the initial problems which had to be resolved for a successful compilation. A benchmark run for PUFF on the REEDA System has been established. The necessary logic has been prepared to modify PUFF to account for:

- a variable atmosphere
- a moving exhaust nozzle
- a solid ground level

A mathematical model for calculating the jet stagnation length has been established. The necessary data for calculating the radiative emittance of the exhaust products have been collected. Once this was accomplished a benchmark comparison was made; however, some discrepancies were noted. When the REEDA version of PUFF was compared to an IBM 370 operational version identical results were obtained from the initial time until time was equal to 0.10 seconds using 0.01 sec time increments. At this point in the execution of the program, the time increment was increased to 0.10 second. Using the new time increment, significant differences begin to appear in the calculated results. Various modifications were made to try to eliminate the difference. All variables and calculations were changed to double precision. Various complex arithmetic computations were re-structured into less compound statements to eliminate possible loss of accuracy by truncation, etc. The above changes have not affected the final results. There still remained differences in the results when the time increment was increased to 0.10 second. Thus it was decided not to increase the time increment, but to leave it constant at .01 second for the entire duration of the program to determine if better accuracy is gained at larger times into the run; however, the results did not change. Consequently, analysis of the REEDA version of the PUFF Program will continue with appropriate modifications being made.

It should be noted, however, that due to the difference in hardware (i.e., IBM 370 vs HP 21MX REEDA), 32 bit vs 16 bit single precision words), 64 bit double precision vs 48 bit double precision words), (16 significant digits vs 11 significant digits), complete agreement between the results of the two machines may not be obtainable given the algorithms that exist currently in the PUFF Program.

6.3 CONCLUSIONS

A new numerical cloud rise program developed for another purpose has been investigated to see if it is suitable for use on the exhaust cloud from the Space Shuttle propulsion system. Development has been initiated to modify the original code and convert it for use on the REEDA system. The use of this code would allow the simultaneous determination of the cloud shape and size and the radiation loss from the exhaust effluents. No other known analyses can handle the situation as aptly as the PUFF code.

7. OVERALL CONCLUSIONS AND RECOMMENDATIONS

The study performed under NASA Contract NAS8-31806 has yielded large dividends in the technology learned, the basic algorithms developed, the meteorological knowledge about KSC brought to a useful form, and the large amount of software developed. New techniques utilizing the touch panel on the Plasmascope have yielded programs that are convenient and rapid to use. The effort has basically completed the necessary homework for a full scale climatological diffusion assessment; what is now required is to bring together the various models and start the development of an operational diffusion model useful not only for a climatological assessment but for monitoring operational launches. The technique for the calculation of the SRB exhaust effluents has been developed but so far losses due to plume impingement, radiation, the flame trench, and water injection have not been considered. A prelininary climatological diffusion assessment was performed to validate the techniques developed; the results have only limited validity and no conclusions can be drawn from the results of the study. The study assumed the Space Shuttle was a Titan type vehicle with only solid propellant boosters; the liquid propellant SSME and their interactions with the solid motor effluents were not considered.

7.1 RECOMMENDED STUDY

Ground-based stable layers and inversions are common over land areas near KSC during calm clear nights, especially in winter The percent frequency of occurrence of ground based inversions for various thickness intervals by season at KSC (50) is given in Table 7-1.

Table 7-1. Percent frequency of occurrence of ground-based inversions by season at KSC during 1965 - 1969 at 0700 and 1900 EST.

Thickness of Ground Based Inversion (m)	Dec Jan Feb	Mar Apr May	June July Aug	Sept Oct Nov
<100	2.1	1.2	1.5	2.1
101 - 250	32.4	25.3	30.2	23.5
251 - 500	23.5	22.8	27.6	26.0
501 - 750	2.5	4.1	1.3	0.6
751 - 1000	1.6	0.6	0	0.6
1000 - 1500	3.5	.2	0	0.2
>1500	0.7	.2	0	0.2

The statistics in Table 7-1 are based on Rawinsonde data. Shallow ground based inversions (thickness less than 250 m) reported at KSC are based on a surface temperature (at 16 ft) and a temperature at the first mandatory pressure level (1,000 mb). Since the temperature at 16 ft is strongly influenced by local micrometeorological conditions the statistics of shallow inversions are not representative of other locations beyond a short distance from the measurement location. However if inversions are reported based on temperature observations at three or more altitudes (including the observation at 16 ft) there is more support for the argument that stable conditions exist near the ground over a wider area in the vicinity of the measurement site. Since a ground based stable layer at a particular location will effectively insulate that location from the stabilized SRB cloud, it is important to establish the applicability of the available KSC inversion statistics to the climatological impact analysis. The physical processes responsible for the formation of ground-based inversions in the areas surrounding KSC are influenced by the relative distribution of rural and urban topography and water bodies. Urban areas and water bodies during winters at KSC represent nocturnal heat sources which could contribute to the maintenance of a nocturnal mixed layer. Although a nocturnal mixed layer has been identified over large cities (51, 52) its existance has not been identified or correlated with nocturnal heat sources in the vicinity of KSC.

It is a reasonable hypothesis that the inversion statistics obtained from KSC Rawinsonde data are not necessarily representative of conditions at all locations of interest in the vicinity of KSC. It is suggested than an experimental study be implemented to establish the relative strength and frequency of occurrence of ground based stable layers and inversions over various locations of interest near KSC. Adequate results would be obtained by sampling temperatures aloft (to 1 km) daily, 1 hour before sunrise, during January and February over population centers (Titusville, Cocoa), working areas, and viewing areas within KSC boundaries. The purpose of the study will be the establishment of the degree of conservatism of air quality impact calculations based on the available large sample of Rawinsonde data at KSC.

8. REFERENCES CITED

1. Stephens, J. B. and Steward, R. B.: Rocket Exhaust Effluent Modeling for Tropospheric Air Quality and Environmental Assessments, Technical Report (To Be Published), Marshall Space Flight Center.

2. Dumbauld, R. K. and Bjorklund, J. R.: NASA/MSFC Multilayer Diffusion Models and Computer Programs - Version 5, NASA CR-2631, Prepared by H. E. Cramer Co. for Marshall Space Flight Center, December 1975.

3. Stephens, J. B. and Sloan, J. C.: Meteorological Regimes for the Classification of Aerospace Air Quality Predictions for NASA-Kennedy Space Center, NASA TMX 3450, Marshall Space Flight Center, October 1976.

4. Stephens, J. B. and Campbell, W. C.: Compendium of Tetroonsonde Measurements Made at Kennedy Space Flight Center During August - September 1975, Technical Memorandum (To Be Published), Marshall Space Flight Center.

5. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meteorological Data for the Titan III C Launch in December 1973, NASA TMX-73334, Marshall Space Flight Cen -, August 1976.

6. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meteorological Data for the Centaur Launch in February 1974, NASA TMX-73335, Marshall Space Flight Center, August 1976.

7. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meteorological Data for the ATS-F Launch in May 1974, NASA TMX-73336, Marshall Space Flight Center, August 1976.

8. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meteorological Data for the Helios A Launch in December 1974, NASA TMX-73337, Marshall Space Flight Center, August 1976.

9. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meteorological Data for the Titan III C Launch in May 1975, NASA TMX-73338, Marshall Space Flight Center, August 1976.

REFERENCES CITED

(Continued)

10. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meterological Data for the Viking A Launch in August 1975, NASA TMX-73339, Marshall Space Flight Center, August 1976.

11. Stephens, J. B., Adelfang, S. I. and Goldford, A. I.: Compendium of Meterological Data for the Viking B Launch in September 1975, NASA TMX-73340, Marshall Space Flight Center, August 1976.

12. Gregory, G. L., Wornom, D. E., Bendura, R. J. and Wagner, H. S.: Hydrogen Chloride Measurements from Titan III Launches at the Air Force Eastern Test Range, 1973 through 1975, NASA TMX-72832, Langley Research Center, March 1976.

13. National Aeronautics and Space Administration Environmental Statement for Space Shuttle, Final, July 1972.

14. Stephens, J. B., Wands, R., and Goldford, A. I.: "Air Quality Guidelines for Aerospace Exhaust Effluents," 1976, Technical Memorandum (To Be Published), Marshall Space Flight Center.

15. Smith, O. E.: Vector Wind and Vector Wind Shear Models 0-27 km Altitude for Cape Kennedy, Florida, and Vandenberg AFB, California, NASA TMX-73319, Marshall Space Flight Center, July 1976.

16. Panofsky, H. A., and Pradsad, B.: Similarity Theories and Diffusion, Int. J. Air Wat. Poll.,(9),pp. 419-430, 1965.

17. Golder, D.: Relations Among Stability Parameters in the Surface Layer, Boundary Layer Meteorology,(3),pp. 45-58, 1972.

18. Blackadar, A. K., Dutton, S. A., Panofsky, H. A. and Chaplin, A.: Investigation of the Turbulent Wind Field Below 150 M Altitude at the Eastern Test Range, NASA CR-1410, Prepared by Pennsylvania State University for Marshall Space Flight Center, August 1969.

19. Dumbauld, R. K., Bjorklund, J. R. and Bowers, J. F.: Handbook for Estimating Toxic Fuel Hazards, NASA CR-61326, Prepared by the H. E. Cramer Co. for Marshall Space Flight Center, 1970.

REFERENCES CITED

(Continued)

20. Personal Communication with Daniel Dembro, Delta Project, NASA Goddard Space Flight Center in Letter Dated 26 January 1976 (NASA File 9602, Reply Attn. of 471).

21. McDonnell Douglas Corporation Transmittal of Data A3-375-DP-B4FO-76-064, 17 August 1976.

22. Personal Communication with J. Lamberty of CSD of United Technologies, 19 December 1974.

23. Anon.: Updated Shuttle Mission 3A Performance Prediction, Teledyne Brown Engineering Technical Letter ASD-EL24-22569, 30 January 1976.

24. H. E. Cramer Co. Computed Values Obtained from Burt Weller at Battelle Memorial Institute, 1970 (Approx. Date).

25. Burns, E. A.: Analysis of Minuteman Exhaust Products, Final Report SRI Project PRD-3753. Stanford Research Institute, Menlo Park, California, October 25, 1962.

26. Personal Communication with Leon Strand of JPL, Pasadena, California, January 9, 1976.

27. Stedman, D. H., Stolarski, R. S., Cicerone, R. J., Williams, J. K.: Rocket Exhaust Effluent Chemistry in the Atmosphere, NASA CR-129042, Prepared by The University of Michigan College of Engineering for Marshall Space Flight Center, March 1974.

28. Dawbarn, R. and Kinslow, M.: Studies of the Exhaust From Solid Propellent Rocket Motors, Arnold Engineering Development Center Report, AEDC-TR-76-49, September, 1976.

29. Willoughby, P. G. and Carlson, R. L.: Sampling and Size Determination of Particles from the Titan III C Exhaust Plume, Technical Report, TR-33-74-U1, United Aircraft Corp., Sunnyvale, California, Februar. 1974.

30. Knutson, E. O., Fenton, D. L.: Atmospheric Scavenging of Hydrochloric Acid, NASA CR-2598, Prepared by IIT Research Institute for Marshall Space Flight Center, August 1975.

REFERENCES CITED

(Continued)

31. Pellett, G. L.: Washout of HCl and Application to Solid Rocket Exhaust Clouds, Paper presented at Precipitation Scavenging Symposium, Champaign, Illinois, October 14-18, 1974.

32. Jex, D. W., Linton, R. C. and Russell, W. M.: The Effects of Solid Rocket Motor Effluents on Selected Surfaces and Solid Particle Size, Distribution, and Composition for Simulated Shuttle Booster Separation Motors, NASA TMX-64975, Marshall Space Flight Center, January 1976.

33. Rhein, R. A.: Hydrochloric Acid Aerosol Formation by the Interaction of Hydrogen Chloride with Humid Air, TM 33-658 Jet Propulsion Laboratory, November 1973.

34. Pieper, J. L.: ICRPG Liquid Propellant Thrust Chamber Performance Evaluation Manual, CPIA Publication No. 178, September 1968.

35. Anon.: JANNAF Rocket Engine Performance Test Data Acquisition and Interpretation Manual, CPIA Publication No. 245, December 1973.

36. Anon.: JANNAF Liquid Rocket Engine Performance, Prediction and Evaluation Manual, CPIA Publication No. 246, December 1973.

37. Goldford, A. I.: Rocket Exhaust Chemistry Studies Relating to the Space Shuttle and Titan III Vehicles, TN-230-1430, Prepared by Northrop Services Inc. for Marshall Space Flight Center, April 1975.

38. Svehla, R. A., and McBride, B. J.: FORTRAN IV Computer Program for Calculation of Thermodynamic and Transport Properties of Complex Chemical Systems, NASA TND-7056, Lewis Research Center, January 1973.

39. Kliegel, J. R. and Nickerson, G. R.: Axisymmetric Two-Phase Perfect Gas Performance Program, 02874-6006-R000, TRW Systems, Redondo Beach, California, April 1967.

40. Goldford, A. I.: Results of SRB Solid Propellent Nozzle Calculations, M9230-74-287, Prepared by Northrop Services, Inc. for Marshall Space Flight Center, May 21, 1974.

REFERENCES CITED

(Continued)

41. Edelman, R., and Weilerstein, G.: Mixing and Combustion in Supersonic Flow with Lateral Pressure Gradient Effects, GASL TR 636, General Applied Science Labs, Westbury, New York, August 1968.

42. Cohen, L. S.: An Analytical Study of the Mixing and Non-equilibrium Chemical Reaction of A Flowing Compressible Streams, AIAA Paper 66-617, presented at the Second Propulsion Joint Specialist Conference, Colorado Springs, Colorado, June 1966.

43. Mikatarian, R. R., and Pergament, H. S.: AeroChem Axisymmetric Mixing with Nonequilibrium Chemistry Computer Program, TP-200, AeroChem Research Laboratories, Inc., Princeton, New Jersey, June 1969.

44. Wojciechowski, C. J. and Penny, M. M.: Development of High Altitude Plume Impingement Analysis for Calculating Heating Rates, Forces and Moments, LMSC-HREC D162867-I, Lockheed Missiles & Space Company, Huntsville, Alabama, March 1971.

45. Hart, W. S.: Dynamics of Large Buoyant Clouds Generated by Rocket Launches, Transactions of the ASME, Journal of Basic Engineering, March 1972, pp. 53-60.

46. Stephens, J. B. and Hamilton, P. A.: Diffusion Algorithms and Data Reduction Routine for Onsite Launch Predictions for the Transport of Titan III C Exhaust Effluents, NASA TN D-7862, Marshall Space Flight Center, December 1974.

47. Stephens, J. B., Ed.: Atmospheric Diffusion Predictions for the Exhaust Effluents from the Launch of a Titan III C, December 13, 1973, NASA TMX-64925, Marshall Space Flight Center, September 1974.

48. Stephens, J. B., Susko, M., Kaufman, J. W. and Hill, C. K.: An Analytical Analysis of the Dispersion Predictions for Effluents from the Saturn V and Scout Algol III Rocket Exhausts, NASA TMX-2935, Marshall Space Flight Center, October 1973.

49. Stephens, J. B.: Diffusion Algorithms and Data Reduction Routines for On-Site Real-Time Launch Predictions for the Transport of Delta-THOR Exhaust Effluents, NASA TND-8194, Marshall Space Flight Center, March 1976.

REFERENCES CITED

(Concluded)

50. Anon.: Inversion Study, Frequency and Percentage Frequency of Temperature Relative Humidity and Wind (Seasonal and Annual) Station 12868 Cape Kennedy Florida, NWRC Job No. 13244, National Oceanographic and Atmospheric Administration, Asheville, North Carolina, April 1976.

51. Clarke, J. F.: Nocturnal Urban Boundary Layer Over Cincinnati, Ohio, Monthly Weather Review, V. 97, No. 8, August 1969, pp 582-589.

52. Davidson, B.: A Summary of the New York Urban Air Pollution Dynamics Research Program, J. Air Pol. Constr. Assoc., Vol. 17, No. 3, March 1967, pp 154-158.

APPENDIX A

SOFTWARE SOURCE LISTINGS

This section contains the complete .ource listings of most of the software programs discussed in this report.

A-1. Conversion Programs (Generic)

● IBM 370/360	BCD/EBCDIC → ASCII
● UNIVAC 1108	BCD/EBCDIC → ASCII
● HP 2100	EBCDIC → ASCII

IBM 370/360 BCD/EBCDIC → ASCII Conversion

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                      1            PRINT NOGEN
                                      2 HMPTAPE   READY
                                     12            RDJFCB (TAPE)
                                     18            OPEN   (CARDIN,,TAPE,(OUTPUT))
00002E                               26 READCARD DS     0H
                                     27            GET    CARDIN
000038 1821                          31            LR     2,1
00003A D503 204C C358 0004C 00358    32            CLC    76(4,2),=C'    U'
000040 4780 C05A      0005A          33            BE     CLOSE
000044 DC4F 2000 C198 00000 00198    34            TR     0(80,2),BCDTOASC
                                     35            PUT    TAPE,(2)
000056 47F0 C02E      0002E          40            B      READCARD
00005A                               41 CLOSE    DS     0H
                                     42            CLOSE (TAPE,LEAVE)
000066 4820 C128      00128          48            LH     2,JFCBAREA+68
00006A 4120 2001      00001          49            LA     2,1(,2)
00006E 4020 C128      00128          50            STH    2,JFCBAREA+68
                                     51            OPEN   (TAPE,(OUTPUT,LEAVE)),TYPE=J
00007E 47F0 C02E      0002E          57            B      READCARD
000082                               58 EOF      DS     0H
                                     59            CLOSE (CARDIN,,TAPE)
                                     67            RET
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

0000E4                               75 JFCBAREA DS    44F
000194 870000E4                      76 LIST     DC    X'87',AL3(JFCBAREA)
000198 0001020304050607              77 BCDTOASC DC    X'000102030405060708090A0B0C0D0E0F'
0001A8 1011121314151617              78          DC    X'101112131415161718191A1B1C1D1E1F'
0001B8 2021222324252627              79          DC    X'202122232425262728292A2B2C2D2E2F'
0001C8 3031323334353637              80          DC    X'303132333435363738393A3B3C3D3E3F'
0001D8 20                            81          DC    X'20'
0001D9 4142434445464748              82          DC    X'4142434445464748494A'
0001E3 2E295B3C4F28                  83          DC    X'2E295B3C4F28'
0001E9 5152535455565758              84          DC    X'5152535455565758595A'
0001F3 242A5D3B5F2D2F                85          DC    X'242A5D3B5F2D2F'
0001FA 6263646566676869              86          DC    X'62636465666768696A'
000203 2C28605C6F                    87          DC    X'2C28605C6F'
000208 7071727374757677              88          DC    X'707172737475767778797A'
000213 3D273A3E7F80                  89          DC    X'3D273A3E7F80'
000219 6162636465666768              90          DC    X'616263646566676869'
000222 8A8B8C8D8E8F90                91          DC    X'8A8B8C8D8E8F90'
000229 6A6B6C6D6E6F7071              92          DC    X'6A6B6C6D6E6F707172'
000232 9A9B9C9D9E9FA0                93          DC    X'9A9B9C9D9E9FA0'
000239 7E73747576777879              94          DC    X'7E737475767778797A'
000242 AAABACADAEAF                  95          DC    X'AAABACADAEAF'
000248 B0B1B2B3B4B5B6B7              96          DC    X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF
000258 3F41424344454647              97          DC    X'3F414243444546474849'
000262 CACBCCCDCECF                  98          DC    X'CACBCCCDCECF'
000268 214A4B4C4D4E4F50              99          DC    X'214A4B4C4D4E4F505152'
000272 DADBDCDDDEDFE0E1             100          DC    X'DADBDCDDDEDFE0E1'
00027A 535455565758595A             101          DC    X'535455565758595A'
000282 EAEBECEDEEEF                 102          DC    X'EAEBECEDEEEF'
000288 3031323334353637             103          DC    X'30313233343536373839'
000292 FAFBFCFDFEFF                 104          DC    X'FAFBFCFDFEFF'


                                    106 CARDIN   DCB   DSORG=PS,MACRF=GL,DDNAME=CARDS,EODAD=EOF,
                                                       RECFM=FB,LRECL=80
                                    160 TAPE     DCB   DSORG=PS,MACRF=PM,DDNAME=TAPE,EXLST=LIST,
                                                       RECFM=FB,LRECL=80
000000                              214          END   HWPTAPE
000358 404040E4                     215                =C'   U'
```

UNIVAC 1108 BCD → ASCII Program

```
@ASM,IS TPF$.SOURCE,TPF$.OBJECT
ASM 14R1-RL72R1 10/06/76  9:22:01 (,0)

   1.                                                     AXR$ .
   2.                                            $(1) .
   3.                                            LISTER .
   4.                                            RDCARD .
   5.      01   000000  10 16 00 00 0 000036             LA,U    A0,CARD          .
   6.           000001  01 01 00 00 0 000014             SA,H2   A0,WORDS         .
   7.           000002  10 16 00 00 0 000035             LA,U    A0,RDPACK        .
   8.  0        000003  72 11 00 00 0 000000             ER      IOW$             .  READ A RECORD.
   9.           000004  10 15 00 00 0 000010             LA,S1   A0,STATUS        .
  10.           000005  53 16 00 00 0 000001             TNE,U   A0,1             .  IS IT AN EOF.
  11.           000006  74 04 00 00 0 000161             J       LASTCD           .  YES.  GO WRITE EOF.
  12.           000007  10 16 00 00 0 000054             LA,U    A0,CARD+14       .
  13.           000010  27 03 01 00 0 000154             LX      A1,(1,LINE)      .
  14.           000011  10 00 01 00 0 000156             LA      A1,(010020040100)  .
  15.           000012  10 00 02 00 0 000156             LA      A2,(020040100200)  .
  16.           000013  10 16 03 00 0 000010             LA,U    A3,8             .
  17.                                            BLANK .
  18.           000014  01 00 01 01 2 000000             SA      A1,0,*X1         .
  19.           000015  01 00 02 01 2 000000             SA      A2,0,*X1         .
  20.           000016  72 017 00 0 000014               JGD     A3,BLANK         .
  21.
  22.           000017  27 00 01 00 0 000157             LX      A1,(1,CARD)      .
  23.           000020  27 00 02 00 0 000154             LX      A2,(1,LINE)      .
  24.                                            LOOP .
  25.           000021  10 15 01 01 0 000000             LA,S1   A1,0,X1          .
  26.           000022  10 00 02 15 0 000054             LA      A2,FD2ASC,A1     .
  27.           000023  73 12 02 00 0 000034             LSSL    A2,28            .
  28.           000024  10 14 01 01 0 000000             LA,S2   A1,0,X1          .
  29.           000025  10 00 16 15 0 000054             LA      A14,FD2ASC,A1    .
  30.           000026  73 12 16 00 0 000024             LSSL    A14,20           .
  31.           000027  40 00 02 00 0 000032             OR      A2,A14           .
  32.           000030  10 13 01 01 0 000000             LA,S3   A1,0,X1          .
  33.           000031  10 00 16 15 0 000054             LA      A14,FD2ASC,A1    .
  34.           000032  73 12 16 00 0 000014             LSSL    A14,12           .
  35.           000033  40 00 03 00 0 000032             OR      A3,A14           .
  36.           000034  10 12 01 01 0 000000             LA,S4   A1,0,X1          .
  37.           000035  10 00 16 15 0 000054             LA      A14,FD2ASC,A1    .
  38.           000036  73 12 16 00 0 000004             LSSL    A14,4            .
  39.           000037  40 00 04 00 0 000032             OR      A4,A14           .
  40.           000040  10 11 01 01 0 000000             LA,S5   A1,0,X1          .
  41.           000041  10 00 16 15 0 000054             LA      A14,FD2ASC,A1    .
  42.           000042  10 16 17 00 0 000000             LA,U    A15,0            .
  43.           000043  73 03 16 00 0 000004             DSL     A14,4            .
  44.           000044  40 00 05 00 0 000032             OR      A5,A14           .
  45.           000045  01 00 06 02 2 000000             SA      A6,0,*X2         .  ODD WORD.
  46.           000046  10 00 02 00 0 000033             LA      A2,A15           .
  47.           000047  10 10 01 01 2 000000             LA,S6   A1,0,*X1         .
  48.           000050  10 00 16 15 0 000054             LA      A14,FD2ASC,A1    .
  49.           000051  73 12 16 00 0 000030             LSSL    A14,24           .
  50.           000052  40 00 02 00 0 000032             OR      A2,A14           .
  51.
  52.
```

```
 53.      000053  06 02 01 00 0 000012              SX,H1    A1,TESTWD         .
 54.      000054  52 02 00 00 0 000012              TE,H1    A0,TESTWD         .
 55.      000055  74 04 00 00 0 000061              J        AR01              .
 56.      000056  40 00 03 00 0 000160              OR       A3,(010020000401) .
 57.      000057  01 00 04 02 0 000000              SA       A4,0,X2           .  EVEN WORD.
 58.      000060  74 04 00 00 0 000154              J        WTLINE            .
 59.                                        .
 60.                                        .
 61.                                        AR01 .
 62.      000061  10 16 01 01 0 000000              LA,S1    A1,0,X1           .
 63.      000062  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
 64.      000063  73 12 16 00 0 000020              LSSL     A14,16            .
 65.      000064  40 00 03 00 0 000032              OR       A3,A14            .
 66.      000065  10 14 01 01 0 000000              LA,S2    A1,0,X1           .
 67.      000066  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
 68.      000067  73 12 16 00 0 000010              LSSL     A14,8             .
 69.      000070  40 00 04 00 0 000032              OR       A4,A14            .
 70.      000071  10 13 01 01 0 000000              LA,S3    A1,0,X1           .
 71.      000072  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
 72.      000073  40 00 05 00 0 000032              OR       A5,A14            .
 73.      000074  01 00 06 02 2 000000              SA       A6,0,*X2          .  EVEN WORD.
 74.      000075  10 12 01 01 0 000000              LA,S4    A1,0,X1           .
 75.      000076  10 00 02 15 0 000054              LA       A2,FD2ASC,A1      .
 76.      000077  73 12 02 00 0 000034              LSSL     A2,28             .
 77.      000100  10 11 01 01 0 000000              LA,S5    A1,0,X1           .
 78.      000101  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
 79.      000102  73 12 16 00 0 000024              LSSL     A14,20            .
 80.      000103  40 00 02 00 0 000032              OR       A2,A14            .
 81.      000104  10 10 01 01 2 000000              LA,S6    A1,0,*X1          .
 82.      000105  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
 83.      000106  73 12 16 00 0 000014              LSSL     A14,12            .
 84.      000107  40 00 03 00 0 000032              OR       A3,A14            .
 85.
 86.                                        .
 87.      000110  06 02 01 00 0 000012              SX,H1    A1,TESTWD         .
 88.      000111  52 02 00 00 0 000012              TE,H1    A0,TESTWD         .
 89.      000112  74 04 00 00 0 000116              J        AR02              .
 90.      000113  40 00 04 00 0 000161              OR       A4,(010402)       .
 91.      000114  01 00 05 02 0 000000              SA       A5,0,X2           .  ODD WORD.
 92.      000115  74 04 00 00 0 000154              J        WTLINE            .
 93.                                        .
 94.                                        .
 95.                                        AR02 .
 96.      000116  10 15 01 01 0 000000              LA,S1    A1,0,X1           .
 97.      000117  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
 98.      000120  73 12 16 00 0 000004              LSSL     A14,4             .
 99.      000121  40 00 04 00 0 000032              OR       A4,A14            .
100.      000122  10 14 01 01 0 000000              LA,S2    A1,0,X1           .
101.      000123  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
102.      000124  10 16 17 00   000000              LA,U     A15,0             .
103.      000125  73 03 16 00 0 000004              DSL      A14,4             .
104.      000126  40 00 05 00 0 000032              OR       A5,A14            .
105.      000127  01 00 06 02 2 000000              SA       A6,0,*X2          .  ODD WORD.
106.      000130  10 00 02 00 0 000033              LA       A2,A15            .
107.      000131  10 13 01 01 0 000000              LA,S3    A1,0,X1           .
108.      000132  10 00 16 15 0 000054              LA       A14,FD2ASC,A1     .
```

```
109.         000133  73 12 16 00 0 000030           LSSL     A19,24          .
110.         000134  40 00 02 00 0 000032           OR       A2,A19          .
111.         000135  10 12 01 01 0 000000           LA,S4    A1,0,X1         .
112.         000136  10 03 16 16 0 000054           LA       A19,FD2ASC,A1   .
113.         000137  73 12 16 00 0 000020           LSSL     A19,16          .
114.         000140  40 00 03 00 0 000032           OR       A3,A19          .
115.         000141  10 11 01 01 0 000000           LA,S5    A1,0,X1         .
116.         000142  10 03 16 15 0 000054           LA       A19,FD2ASC,A1   .
117.         000143  73 12 16 00 0 000010           LSSL     A19,8           .
118.         000144  40 00 04 00 0 000032           OR       A4,A19          .
119.         000145  10 10 01 01 2 000000           LA,S6    A1,0,*X1        .
120.         000146  10 03 16 15 0 000054           LA       A19,FD2ASC,A1   .
121.         000147  40 00 05 00 0 000032           OR       A5,A19          .
122.         000150  01 03 06 02 2 000000           SA       A6,0,*X2        .  EVEN WORD.
123.
124.                                       .
125.         000151  06 02 01 00 0 000012           SX,H1    A1,TESTWD       .
126.         000152  52 02 00 00 0 000012           TE,H1    A0,TESTWD       .
127.         000153  74 04 00 00 0 000021           J        LOOP            .
128.                                       .
129.                                       WTLINE .
130.         000154  10 16 00 00 000013             LA,U     A0,LINE         .
131.         000155  01 01 00 00 0 000004           SA,H2    A0,WORD4        .
132.         000156  10 16 00 00 000000             LA,U     A0,TAPEPK       .
133. U       000157  72 11 00 00 0 000000           ER       IO$             .  WRITE A RECORD.
134.         000160  74 04 00 00 0 000056           J        RDCARD          .
135.                                       .
136.                                       LASTCD .
137.         000161  10 16 00 00 000011             LA,U     A0,011          .  WEOF FUNCTION CODE.
138.         000162  01 14 00 00 0 000003           SA,S2    A0,FUN          .
139.         000163  10 16 00 00 000000             LA,U     A0,TAPEPK       .
140. U       000164  72 11 00 00 0 000000           ER       IO$             .  WEOF.
141.         000165  10 16 00 00 000000             LA,U     A0,TAPEPK       .
142. U       000166  72 11 00 00 0 000000           ER       IO$             .  WEOF.
143. U       000167  72 11 00 00 0 000000           ER       EXIT$           .
144.                                       / .
145.                                       $(0) .
146.     00  000000  223631062512          TAPEPK   .        'MYTAPE'        .
147.         000001  050505050505                   .        '      '        .
148.         000002  000000000000                   .        0               .
149.         000003  001000000000          FUN      .        001000000000    .
150.         000004  000022 000013         WORD4    .        022,LINE        .
151.                                       .
152.         000005  310625126705          RDPACK   .        'TAPE7 '        .
153.         000006  050505050505                   .        '      '        .
154.         000007  000000000000                   .        0               .
155.         000010  002000000000          STATUS   .        002000000000    .
156.         000011  000016 000036         RWORD4   .        016,CARD        .
157.                                       .
158.         000012  000000000000          TESTWD   .        0               .
159.         000013                        LINE     RES      19              .
160.         000036                        CARD     RES      14              .
```

```
161.                                              / .
162.        00    000054  000000000100            FD2ASC    +      0100 .         AT SIGN,
163.              000055  000000000133                      +      0133 .         SQUARE BRAK OPEN
164.              000056  000000000135                      +      0135 .         SQUARE BRAK CLOSE
165.              000057  000000000043                      +      0043 .         POUND SIGN
166.              000060  000000000134                      +      0134 .
167.              000061  000000000040                      +      0040 .         SPACE
168.              000062  000000000101                      +      0101 .         A
169.              000063  000000000102                      +      0102 .         B
170.              000064  000000000103                      +      0103 .         C
171.              000065  000000000104                      +      0104 .         D
172.              000066  000000000105                      +      0105 .         E
173.              000067  000000000106                      +      0106 .         F
174.              000070  000000000107                      +      0107 .         G
175.              000071  000000000110                      +      0110 .         H
176.              000072  000000000111                      +      0111 .         I
177.              000073  000000000112                      +      0112 .         J
178.              000074  000000000113                      +      0113 .         K
179.              000075  000000000114                      +      0114 .         L
180.              000076  000000000115                      +      0115 .         M
181.              000077  000000000116                      +      0116 .         N
182.              000100  000000000117                      +      0117 .         O
183.              000101  000000000120                      +      0120 .         P
184.              000102  000000000121                      +      0121 .         Q
185.              000103  000000000122                      +      0122 .         R
186.              000104  000000000123                      +      0123 .         S
187.              000105  000000000124                      +      0124 .         T
188.              000106  000000000125                      +      0125 .         U
189.              000107  000000000126                      +      0126 .         V
190.              000110  000000000127                      +      0127 .         W
191.              000111  000000000130                      +      0130 .         X
192.              000112  000000000131                      +      0131 .         Y
193.              000113  000000000132                      +      0132 .         Z
194.              000114  000000000051                      +      0051 .         PRAN CLOSE
195.              000115  000000000055                      +      0055 ,         MINUS
196.              000116  000000000053                      +      0053 .         PLUS
197.              000117  000000000074                      +      0074 .         LESS THAN
198.              000120  000000000075                      +      0075 .         EQUAL SIGN
199.              000121  000000000076                      +      0076 .         GREATER THAN
200.              000122  000000000046                      +      0046 .         AMPERSAND
201.              000123  000000000044                      +      0044 .         $
202.              000124  000000000052                      +      0052 .         *
203.              000125  000000000050                      +      0050 .         PRAN OPEN
204.              000126  000000000045                      +      0045 .         PERCENT SIGN
205.              000127  000000000072                      +      0072 .         COLON
206.              000130  000000000077                      +      0077 .         QUESTION MARK
207.              000131  000000000041                      +      0041 .         BANG CHAR
208.              000132  000000000054                      +      0054 .         COMMA
209.              000133  000000000134                      +      0134 .         BACK SLASH
210.              000134  000000000060                      +      0060 .         0
211.              000135  000000000061                      +      0061 .         1
212.              000136  000000000062                      +      0062 .         2
213.              000137  000000000063                      +      0063 .         3
214.              000140  000000000064                      +      0064 .         4
215.              000141  000000000065                      +      0065 .         5
216.              000142  000000000066                      +      0066 .         6
```

```
217.          000143  000000000067                       +        0067 .          7
218.          000144  000000000070                       +        0070 .          8
219.          000145  000000000071                       +        0071 .          9
220.          000146  000000000047                       +        0047 .          QUOTE MARK
221.          000147  000000000073                       +        0073 .          SEMICOLON
222.          000150  000000000057                       +        0057 .          SLASH
223.          000151  000000000056                       +        0056 .          PERIOD
224.          000152  000000000042                       +        0042 .          DOUBLE QUOTE
225.          000153  000000000137                       +        0137 .
226.                  000000000000                       END      LISTEN
              000154  000001 000013
              000155  100200401002
              000156  004010020040
              000157  000001 000036
              000160  000010020040
              000161  000000001002

UNDEFINED SYMBOLS:
EXIT*       10RS

END ASM, ERRORS : NONE
```

HP 2100 EBCDIC → ASCII Conversion Programs

```
LIST . T=00004 IS ON CR00011 USING 00013 BLKS R=0073

ASMB,R,L,F
*
*  THIS SUBROUTINE CONVERTS REAL NUMBERS IN IBM FORMAT TO HP FORMAT
*
*  THE CALLING SEQUENCE IS: CALL IBMHP(IBM1,IBM2,HP)
*         WHERE: IBM1 - THE MOST SIGNIFICANT PART OF THE IBM REAL WORD
*                IBM1 - THE LEAST SIGNIFICANT PART OF THE IBM REAL WORD
*                HP - THE REAL WORD IN WHICH THE RESULT IS TO BE STORED
*                       IN HP FORMAT
*
       NAM IBMHP
       ENT IBMHP
       EXT .ENTR
*
IBM1   BSS 1
IBM2   BSS 1
HP1    BSS 1
*
IBMHP  NOP              ENTRY/EXIT POINT
*
       JSB .ENTR        GET ADDRESSES OF PARAMETERS INTO
       DEF IBM1          IBM1, IBM2, AND HP1
*
       LDA HP1          SET UP THE ADDRESS OF THE SECOND
       ISZ A             HALF OF THE HP REAL WORD
       STA HP2
*
       LDA IBM1,I       GET FIRST PART OF IBM REAL WORD
       AND =B077400     MASK OFF THE EXPONENT
       ALF,ALF          SHIFT TO LOWER 8 BITS OF A
       ADA IBIAS        REMOVE IBM EXPONENT BIAS OF 64
       ADA A            QUADRUPLE EXPONENT TO MAKE IT
       ADA A             A POWER OF 2
       SSA              SKIP IF POSITIVE EXPONENT
       JMP NEGEX        JUMP IF NEGATIVE EXPONENT
       ALR              POSITIVE EXPONENT -- ROTATE IT
       JMP STEX          1 BIT RIGHT AND STORE
NEGEX  RAL              NEGATIVE EXPONENT -- SHIFT IT 1
       AND =B000376      MASK OFF JUST THE EXPONENT
       SZA               (SKIP IF EXACTLY ZERO)
       IOR =B000001      PUT IN SIGN BIT
STEX   STA TEMP         STORE THE EXPONENT TEMPORARILY
*
       LDA IBM1,I       GET FIRST PART OF IBM REAL WORD
       RAL              ROTATE IT LEFT 1 BIT
       AND =B000001     MASK OFF MANTISSA SIGN BIT
       STA B            STORE IN B REGISTER
*
       LDA IBM2,I       GET SECOND PART OF IBM REAL WORD
       ARS              DROP LEAST SIGNIFICANT BIT OF MANTISSA
       AND =B077777      AND CLEAR THE UPPER BIT
       SZB,RSS          SKIP IF MANTISSA IS NEGATIVE
       JMP *+2          DONT COMPLEMENT IF MANTISSA POSITIVE
       CMA,INA          MANTISSA NEGATIVE, COMPLEMENT
       AND =B000377     GET LOWEST EIGHT BITS
       ALF,ALF          PUT IN UPPER PART OF WORD
       IOR TEMP         OR IN THE EXPONENT
       STA HP2,I        PUT IN SECOND HALF OF HP REAL WORD
```

```
*
      LDA IBM2,I   GET SECOND PART OF IBM REAL WORD
      AND =B177400 MASK OFF UPPER 8 BITS
      ALF,ALF      ROTATE TO LOWER 8 BITS
      STA TEMP     STORE TEMPORARILY
      LDA IBM1,I   GET FIRST PART OF IBM REAL WORD
      AND =B000377 MASK OFF LOWER 8 BITS
      ALF,ALF      ROTATE TO UPPER 8 BITS
      IOR TEMP     OR IN THE OTHER PART OF MANTISSA
      ARS          GET RID OF THE LOWER BIT
      AND =B077777  AND CLEAR THE UPPER BIT
      SZB,RSS      SKIP IF THE MANTISSA IS NEGATIVE
      JMP *+2      DONT COMPLEMENT IF MANTISSA POSITIVE
      CMA          COMPLEMENT UPPER PART OF MANTISSA
      STA HP1,I    PUT IN FIRST HALF OF HP REAL WORD
*
      JMP IBMHP,I  RETURN TO THE CALLING PROGRAM
*
*
A     EQU 0
B     EQU 1
IBIAS DEC -64
TEMP  NOP
HP2   BSS 1
      END
```

```
FTN4,L
      SUBROUTINE E2A(IA,LIA)
      INTEGER EBCASC(256)
      DIMENSION IA(1)
      DATA EBCASC/0,1,2,3,0,9,0,127,0,0,0,11,12,13,14,15,16,17,18,0,
     .           0,0,8,0,24,25,0,0,0,0,0,0,0,0,28,0,0,0,23,27,
     .           0,0,0,0,0,5,6,7,0,0,22,0,0,30,0,4,0,0,0,0,
     .           20,21,0,26,32,0,0,0,0,0,0,0,0,0,46,60,40,43,0,
     .           38,0,0,0,0,0,0,0,0,0,33,36,42,41,59,94,45,47,0,0,
     .           0,0,0,0,0,0,124,44,37,95,62,63,0,0,0,0,0,0,0,0,
     .           0,96,58,35,64,39,61,34,0,97,
     .           98,99,100,101,102,103,104,105,0,0,
     .           0,0,0,0,0,106,107,108,109,110,
     .           111,112,113,114,0,0,0,0,0,0,
     .           0,126,115,116,117,118,119,120,121,122,
     .           0,0,0,91,0,0,0,0,0,0,
     .           0,0,0,0,0,0,0,0,0,93,
     .           0,0,123,65,66,67,68,69,70,71,
     .           72,73,0,0,0,0,0,0,125.74,
     .           75,76,77,78,79,80,81,82,0,0,
     .           0,0,0,0,92,0,83,84,85,86,
     .           87,88,89,90,0,0,0,0,0,0,
     .           48,49,50,51,52,53,54,55,56,57,
     .           0,0,0,0,0,0/
      DO 7 I=1,LIA
      INDEX1 = IAND(ISHIF(IA(I),-8),177B) + 1
      INDEX2 = IAND(IA(I),177B) + 1
    7 IA(I) = IOR(ISHIF(EBCASC(INDEX1),8),EBCASC(INDEX2))
      RETURN
      END
      END$
```

A-2. Conversion Programs (Specific)

- 1965 KSC Rawinsonde Data Conversion Program
- 1974 Vandenberg Rawinsonde Data Conversion Program
- 1964-1970 Jimsphere Data Conversion Programs

1965 KSC Rawinsonde Data Conversion Program

```
FTN4,L
      PROGRAM SOUND
      DIMENSION IBUF(40),OBUF(40),ISIZE(2)
      DIMENSION NAME(3), IDCB(256)
      INTEGER OBUF
      DATA I99/2H 9/
      DATA ISIZE/-1,40/
      DATA IST/2HST/
      DATA NAME/2H&S,2HDB,2H65/
C** CREATE DISC FILE TO STORE CONVERTED SOUNDING
      CALL CREAT(IDCB,IERR,NAME,ISIZE,3)
C     SET IL TO NUMBER OF WORDS TO BE WRITTEN = 39
      IL=39
      NC = 1
      WRITE(6,320)
56    READ(8,15) OBUF
C     IF(OBUF(2).NE.IST) GO TO 56
C     NC=NC +1
C     IF(NC.LE.12) GO TO 56
      WRITE(6,310) (OBUF(N),N=1,40)
      CALL CODE
      WRITE (IBUF,15) (OBUF(N),N=1,40)
      CALL WRITF (IDCB,IERR,IBUF,IL)
88    DO 10 I=1,5
      NC=NC+1
      READ(8,15) OBUF
15    FORMAT(40A2)
      WRITE(6,310) (OBUF(N),N=1,40)
      CALL CODE
      WRITE (IBUF,15) (OBUF(N),N=1,40)
      CALL WRITF (IDCB,IERR,IBUF,IL)
10    CONTINUE
16    READ(8,20) IALT,IWD,IWKTS,ITEMP,ITS,IDPT,IDS,IPRESS,IRH,IAB,
     1IDEN,IR,IVS,IWS
20    FORMAT(I6,3X,I3,2X,I3,3X,I2,A2,4X,I2,A2,3X,I5,3X,I2,4X,I4,1X,
     1I5,3X,I3,2X,I3,2X,I3,5X)
      CALL ISIGC(ITEMP,ITS,IDPT,IDS,TEMP,DPT)
      PRESS = IPRESS/10.
      AB = IAB/100.
      DEN = IDEN/10.
      WRITE(6,350) IALT,IWD,IWKTS,TEMP,DPT,PRESS,IRH,AB,DEN,IR,IVS,I
350   FORMAT(1X,I6,3X,I3,5X,I3,2X,F5.1,3X,F5.1,3X,F6.1,3X,I2,3X,
     1F5.2,2X,F6.1,1X,I3,1X,I3,1X,I3)
      CALL CODE
      WRITE(IBUF,350) IALT,IWD,IWKTS,TEMP,DPT,PRESS,IRH,AB,DEN,
     1 IR,IUS,IWS
      CALL WRITF (IDCB,IERR,IBUF,IL)
      IF(IALT.LE.19500) GO TO 16
17    READ(8,15) OBUF
      IF(OBUF(40).NE.I99) GO TO 17
      WRITE(6,310) (OBUF.N),N=1,40)
      CALL CODE
      WRITE (IBUF,15) (OBUF(N),N=1,40)
      CALL WRITF (IDCB,IERR,IBUF,IL)
      WRITE(6,320)
      DO 18 I=1,2
      READ(8,15) OBUF
      WRITE(6,310) (OBUF(N),N=1,40)
      CALL CODE
      WRITE (IBUF,15) (OBUF(N),N=1,40)
      CALL WRITF (IDCB,IERR,IBUF,IL)
18    CONTINUE
21    READ(8,19) IALT,IWD,IWKTS,ITEMP,ITS,IDPT,IDS,IPRESS,IRH
19    FORMAT(I6,3X,I3,2X,I3,3X,I2,A2,4X,I2,A2,3X,I5,3X,I2)
      PRESS = IPRESS/10.
      CALL ISIGC(ITEMP,ITS,IDPT,IDS,TEMP,DPT)
      WRITE(6,351) IALT,IWD,IWKTS,TEMP,DPT,PRESS,IRH
351   FORMAT(1X,I6,3X,I3,5X,I3,2X,F5.1,3X,F5.1,3X,F6.1,3X,I2)
      CALL CODE
      WRITE(IBUF,351) IALT,IWD,IWKTS,TEMP,DPT,PRESS,IRH
      CALL WRITF (IDCB,IERR,IBUF,IL)
      IF(IALT.LE.19500) GO TO 21
86    READ(8,15) OBUF
      IF(OBUF(2).NE.IST) GO TO 86
      WRITE(6,320)
      WRITE(6,310) (OBUF(N),N=1,40)
      CALL CODE
      WRITE (IBUF,15) (OBUF(N),N=1,40)
      CALL WRITF (IDCB,IERR,IBUF,IL)
      IF(NC.GT.700 )GO TO 90
      GO TO 88
320   FORMAT(1H1)
310   FORMAT(1X,40A2)
90    CALL CLOSE(IDCB,IERR)
      END
      SUBROUTINE ISIGC(ITEMP,ITS,IDPT,IDS,TEMP,DPT)
      DIMENSION ICHR(10), INM(10)
      DATA ICHR/2H! ,2HJ ,2HK ,2HL ,2HM ,2HN ,2HO ,2HP ,2HQ ,2HR /
      DATA INM/2H0 ,2H1 ,2H2 ,2H3 ,2H4 ,2H5 ,2H6 ,2H7 ,2H8 ,2H9 /
      DO 10 I=1,10
      A = I - 1
      B = -1.
      IF(ITS.EQ.ICHR(I)) TEMP =(FLOAT(ITEMP) + A/10.) * B
      IF(IDS.EQ.ICHR(I)) DPT =(FLOAT(IDPT) + A/10.) * B
      IF(ITS.EQ.INM(I)) TEMP =(FLOAT(ITEMP) + A/10.)
      IF(IDS.EQ.INM(I)) DPT =(FLOAT(IDPT) + A/10.)
10    CONTINUE
      RETURN
      END
      END$
```

1974 Vandenberg Rawinsonde Data Conversion Programs

- IBM 360/44 Variable Length → Fixed Length IBM 370
- Data Selection Program
- IBM 370 EBCDIC → ASCII (See pages A-3 through A-5)

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                      1            PRINT NOGEN
                                      2 MPRTRD     READY
000014 47F0 C09E      0009E          12 BRANCH     B     OPEN
000018 5820 1000      00000          13            L     2,0(,1)
00001C 5830 1004      00004          14            L     3,4(,1)
                                     15            GET   TAPE,(2)
00002C 1B11                          20            SR    1,1
00002E 5010 3000      00000          21            ST    1,0(,3)
000032                               22 RETURN     DS    0H
                                     23            RET


000084                               31 EOF        DS    0H
000084 96F0 C015      00015          32            OI    BRANCH+1,X'F0'
                                     33            CLOSE (TAPE)
000092 4110 0001      00001          39            LA    1,1
000096 5010 3000      00000          40            ST    1,0(,3)
00009A 47F0 C032      00032          41            B     RETURN


00009E                               43 OPEN       DS    0H
00009E 1821                          44            LR    2,1
0000A0 940F C015      00015          45            NI    BRANCH+1,X'0F'
                                     46            OPEN  (TAPE)
0000AE 1812                          52            LR    1,2
0000B0 47F0 C018      00018          53            B     BRANCH+4


                                     55 TAPE       DCB   DSORG=PS,MACRF=GM,DDNAME=SASTAPE,EODAD=EOF
                                    109            END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=54,SIZE=0200K,
                          SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002              DIMENSION IMN(12)
ISN 0003              LOGICAL*1 IDATA(20),IBUF(2500)
ISN 0004              DIMENSION PRESSP(78),PRESSR(78),ALTP(78),ALTR(78),TEMPP(78),
                     1          TEMPR(78),DPTP(78),DPTR(78),WDP(78),WDR(78),WSP(78),
                     1          WSR(78)
ISN 0005              INTEGER*2 YR,MO,DA,HR,OB,IND,PRESS,TEMP,DPT,WD,WS,NLEVEL
ISN 0006              INTEGER  HT,STN,EOF
ISN 0007              EQUIVALENCE  (IBUF(5),NLEVEL),(IBUF(9),STN), (IBUF(13),YR),
                     1              (IBUF(15),MO), (IBUF(17),DA), (IBUF(19),HR),
                     1              (IBUF(25),OB), (IDATA(1),IND), (IDATA(3),PRESS),
                     1              (IDATA(5),HT), (IDATA(9),TEMP),(IDATA(11),DPT),
                     1              (IDATA(13),WD), (IDATA(15),WS)
ISN 0008              DATA IMN/4HJAN ,4HFEB ,4HMAR ,4HAPR ,4HMAY ,4HJUN ,4HJUL ,4HAUG ,
                     14HSEP ,4HOCT ,4HNOV ,4HDEC /
                C  ** CALL TAPE READ ROUTINE
ISN 0009              ICNT = 0
ISN 0010         10   CALL MPRTR3(IBUF,EOF)
ISN 0011              IF(EOF.EQ.1) GO TO 99
                C ** CHECK FOR PIBAL LEVEL
                C ** CHECK FOR RAWINSONDE DATA
ISN 0013              IF(OB.LT.1.OR.OB.GT.2) GO TO 10
                C ** CHECK FOR SURFACE LEVEL
ISN 0015              DO 15 I=1,20
ISN 0016              N = I + 80
ISN 0017              IDATA(I) = IBUF(N)
ISN 0018         15   CONTINUE
ISN 0019              IF(IND.NE.0) GO TO 10
                C  ** COMPUTE NUMBER OF DATA LEVELS
ISN 0021              ILEVEL = NLEVEL
ISN 0022              LEVELS = ILEVEL/20
ISN 0023              IF(OB.EQ.1) LEVELP = LEVELS
ISN 0025              IF(OB.EQ.2) LEVELR = LEVELS
                C ** PROCESS ALL LEVELS OF DATA
ISN 0027              DO 20 I=1,LEVELS
ISN 0028              II = I*20 + 61
ISN 0029              JJ = II + 19
ISN 0030              K = 0
ISN 0031              DO 25 J=II,JJ
ISN 0032              K = K + 1
ISN 0033              IDATA(K) = IBUF(J)
ISN 0034         25   CONTINUE
ISN 0035              IF(OB.EQ.2) GO TO 88
ISN 0037              PRESSP(I) = PRESS
ISN 0038              PRESSP(I) = PRESSP(I)/10.
ISN 0039              ALTP(I) = HT
ISN 0040              TEMPP(I) = TEMP
ISN 0041              TEMPP(I) = (TEMPP(I)/10.) - 273.16
ISN 0042              WDP(I) = WD
ISN 0043              WSP(I) = WS/10.
ISN 0044              DPTP(I) = DPT
ISN 0045            IDAP = DA
ISN 0046            IHRP = HR
ISN 0047            GO TO 20
ISN 0048       88   CONTINUE
ISN 0049            PRESSR(I) = PRESS
ISN 0050            PRESSR(I) = PRESSR(I)/10.
ISN 0051            ALTR(I) = HT
ISN 0052            TEMPR(I) = TEMP
ISN 0053            TEMPR(I) = (TEMPR(I)/10.) - 273.16
ISN 0054            WDR(I) = WD
ISN 0055            WSR(I) = WS/10.
ISN 0056            DPTR(I) = DPT
ISN 0057            IDAR = DA
ISN 0058            IHRR = HR
ISN 0059       20   CONTINUE
ISN 0060            ISTN = STN
ISN 0061            IYR = YR
ISN 0062            IMO = MO
ISN 0063            IDA = DA
ISN 0064            IHR = HR
               C  ** WRITE DATA RECORDS TO TAPE
ISN 0065            IF(OB.EQ.1) GO TO 10
ISN 0067            IF(OB.NE.2) GO TO 10
ISN 0069            ICNT = ICNT + 1
ISN 0070            IF(IDAP.NE.IDAR.OR.IHRP.NE.IHRR) GO TO 10
ISN 0072            IF(IHRP.NE.12) GO TO 10
ISN 0074            IF(ICNT.LT.11) GO TO 10
ISN 0076            ICNT = 0
               C ** COMPUTE PIBAL PRESSURES
ISN 0077            IF(PRESSP(1).EQ.-.1.AND.ALTP(1).EQ.ALTR(1)) PRESSP(1) = PRESSR(1)
ISN 0079            N = 1
ISN 0080            DO 210 I=2,LEVELP
ISN 0081            IF(PRESSP(I).NE.-.1) GO TO 220
ISN 0083            N = N + 1
ISN 0084            GO TO 210
ISN 0085      220   IF(N.EQ.1) GO TO 210
ISN 0087            J = I - N
ISN 0088            X = ALTP(I) - ALTP(J)
ISN 0089            Y = ALOG(PRESSP(J))
ISN 0090            Z = ALOG(PRESSP(I))
ISN 0091            C = Y - Z
ISN 0092            NN = N-1
ISN 0093            DO 32 K=1,NN
ISN 0094            A = ALTP(J+K) - ALTP(J)
ISN 0095            B = A/X
ISN 0096            D = B*C
ISN 0097            E = Y - D
ISN 0098            PRESSP(J+K) = EXP(E)
ISN 0099      32    CONTINUE
ISN 0100            N = 1
ISN 0101        210 CONTINUE
               C ** COMPUTE RADB ALTITIUDES
```

```
ISN 0102              N = 1
ISN 0103              DO 320 I =1,LEVELR
ISN 0104              IF(ALTR(I).NE.-1.) GO TO 330
ISN 0106              N = N + 1
ISN 0107              GO TO 320
ISN 0108       330    IF(N.EQ.1) GO TO 320
ISN 0110              J = I - N
ISN 0111              X = ALOG(PRESSR(J))
ISN 0112              Y = ALOG(PRESSR(I))
ISN 0113              Z = X - Y
ISN 0114              A = (ALTR(I) - ALTR(J))
ISN 0115              NN = N - 1
ISN 0116              DO 33 K = 1,NN
ISN 0117              B = X - ALOG(PRESSR(J+K))
ISN 0118              C = B/Z
ISN 0119              D = A * C
ISN 0120              ALTR(J+K) = D+ ALTR(J)
ISN 0121       33     CONTINUE
ISN 0122              N = 1
ISN 0123       320    CONTINUE
               C ** COMPUTE PIBAL TEMPERATURES
ISN 0124              IF(ALTP(1).EQ.ALTR(1)) TEMPP(1) = TEMPR(1)
ISN 0126              DO 264 I =2,LEVELP
ISN 0127              DO 263 K=2,LEVELR
ISN 0128              IF(ALTP(I).EQ.ALTR(K)) TEMPP(I) = TEMPR(K)
ISN 0130              IF(ALTP(I).GT.3500.) GO TO 265
ISN 0132              IF(ALTP(I).EQ.ALTR(K)) GO TO 264
ISN 0134              IF(ALTP(I).GT.ALTR(K)) GO TO 263
ISN 0136              A = ALTR(K) - ALTR(K-1)
ISN 0137              B = ALTP(I) - ALTR(K-1)
ISN 0138              C = B/A
ISN 0139              D = TEMPR(K-1) - TEMPR(K)
ISN 0140              E = D*C
ISN 0141              TEMPP(I) = TEMPR(K-1) - E
ISN 0142              GO TO 264
ISN 0143       263    CONTINUE
ISN 0144       264    CONTINUE
ISN 0145       265    CONTINUE
ISN 0146              LPN = I - 1
ISN 0147              WRITE(6,421)
ISN 0148       421    FORMAT(//,1H ,17HINTERPOLATED DATA,//)
ISN 0149              WRITE(8,801)
ISN 0150       801    FORMAT(27HTEST NBR 03717 04834 0-24HR)
ISN 0151              WRITE(8,802)
ISN 0152       802    FORMAT(20HRAWINSONDE-PIBAL RUN)
ISN 0153              WRITE(8,803)
ISN 0154       803    FORMAT(21HVANDENBERG AFB, CALIF)
ISN 0155              WRITE(8,804) IDA,IMN(IMO), IYR
ISN 0156       804    FORMAT(7H1200Z  ,I2,1X,A4,2H19,I2)
ISN 0157              WRITE(6,931) IDA,IMN(IMO), IYR
ISN 0158       931    FORMAT(1H ,I2,1X,A4,1X,I2)
ISN 0159              WRITE(8,805)
ISN 0160       805    FORMAT(11HASCENT NBR )
ISN 0161              WRITE(8,701)
ISN 0162              WRITE(8,702)
ISN 0163              WRITE(6,701)
ISN 0164       701    FORMAT(2X,31HALT  DIR SPD  TEMP  DPT   PRESS)
ISN 0165              WRITE(6,702)
ISN 0166       702    FORMAT(3X,29HFT  DEG KTS DEG C         MBS)
ISN 0167              DO 779 I=1,LPN
ISN 0168              IALTN = ALTP(I) * 3.28084
ISN 0169              IWDN = WDP(I)
ISN 0170              IWSN = WSP(I) * 1.94254
ISN 0171              DPTN = DPTP(I)
ISN 0172              TEMPN = TEMPP(I)
ISN 0173              PRESSN = PRESSP(I)
ISN 0174              WRITE(8,703) IALTN,IWDN,IWSN,TEMPN,DPTN,PRESSN
ISN 0175              WRITE(6,703) IALTN,IWDN,IWSN,TEMPN,DPTN,PRESSN
ISN 0176       703    FORMAT(I6,1X,I3,1X,I3,1X,F5.1,1X,F5.1,1X,F7.2)
ISN 0177       779    CONTINUE
ISN 0178              GO TO 10
ISN 0179       99     STOP
ISN 0180              END
```

1964 - 1970 Jimsphere Conversion Programs

- UNIVAC 1108 Data Reduction
- UNIVAC 1108 BCD → ASCII (See pages A-6 through A-11)

```
00000   *DIAGNOSTIC*  THE NAME A APPEARS IN A DIMENSION OR TYPE STATEMENT BUT IS NEVER REFERENCED.
00101        1.            DIMENSION IDATA(300),WD19(19)                                      000000
00103        2.            DIMENSION A(8)                                                     000001
00104        3.            READ(5,1000) NFILES                                                000001
00107        4.            DO 50 NF=1,NFILES                                                  000007
00112        5.            WRITE(6,1050) NF                                                   000023
00115        6.            DO 40 NR=1,20                                                      000033
00120        7.            CALL NTRAN(8,2,298,IDATA,IERR,22)                                  000033
00121        8.     10     IF(IERR.EQ.-1) GO TO 10                                            000043
00123        9.            IF(IERR.LT.-1) CALL NTRAN(8,22)                                    000045
00125       10.            NDX = 16                                                           000055
00126       11.            DO 30 I=1,5                                                        000061
00131       12.            ENCODE (80,1030,WD19,INUM) (IDATA(NDX+7*K),K=1,8)                  000063
00137       13.            NDX=NDX+56                                                         000100
00140       14.            CALL NTRAN(9,1,19,WD19,IERR,22)                                    000103
00141       15.            IF(NF.GE.2) GO TO 25                                               000113
00143       16.            WRITE(6,1090) (WD19(K),K=1,19)                                     000115
00146       17.   1090     FORMAT(1H ,19A6)                                                   000127
00147       18.     25     CONTINUE                                                           000127
00150       19.     30     CONTINUE                                                           000127
00152       20.     40     CONTINUE                                                           000127
00154       21.            CALL NTRAN(8,7,1,22)                                               000127
00155       22.     50     CONTINUE                                                           000137
00157       23.            CALL NTRAN(9,9)                                                    000137
00160       24.            CALL NTRAN(9,11)                                                   000143
00161       25.            STOP                                                               000147
00162       26.   1000     FORMAT(I5)                                                         000153
00163       27.   1030     FORMAT(8F10.1)                                                     000153
00164       28.   1050     FORMAT(1H ,3HNF=,I5)                                               000153
00165       29.            END                                                                000153

     END OF COMPILATION:          1 DIAGNOSTICS.
```

A-3. Interactive REEDA Programs

- MOD3A
- MOD3B
- METPL REED Program*
- STAN5
- MIXH
- JWSPL
- JWDPL
- JIMPS
- SKEW T (Version I & II)
- PUFF

*These have been merged into the NASA/MSFC REED Diffusion Model Program Version I.

Program MOD3A

```
FTN4,L
      PROGRAM MOD3A
C
C**************************************************************************
C
C          NASA/MSFC MULTILAYER DIFFUSION MODEL - MOD3A    04 MAY 1977
C
C**************************************************************************
C
C
C          COMMON BLOCK
C
      COMMON ALT(31), L1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C          INPUT FORMAT STATEMENTS
C
  100 FORMAT (I2,1X,2A2,I2)
  101 FORMAT (10A2)
  102 FORMAT (I4,5XI2,1XA2,A1,1XI4)
  103 FORMAT (I4,3XI2,1XA2,A1,1XI4)
  104 FORMAT (I6,1XI3,1XF4.1,F6.1,F6.1,F7.2,11XF7.2)
C
C          OUTPUT FORMAT STATEMENTS
C
  200 FORMAT (////"Ɛ&dB****NASA/MSFC MULTILAYER DIFFUSION MODEL - MOD3A"
     .        4X"04 MAY 1977****")
  201 FORMAT (//"Ɛ&dBNUMBER OF RUNS AND COMMON DATA FILE NAME "
     .       "(e.g. 01,DATA): Ɛ&dJ_")
  202 FORMAT (5X"RUN "I2" WILL USE DATA FILE "3A2)
  203 FORMAT (//"Ɛ&dFRƐ&dBESEARCH OR Ɛ&dFPƐ&dBRODUCTION RUN: Ɛ&dJ_")
  204 FORMAT (5X"RESEARCH RUN")
  205 FORMAT (5X"PRODUCTION RUN")
  206 FORMAT (//"Ɛ&dBTOP OF SURFACE LAYER(M): Ɛ&dJ_")
  207 FORMAT (//"Ɛ&dBSIGMA OF WIND AZIMUTH ANGLE: Ɛ&dJ_")
  208 FORMAT (//"Ȩ&dBLAUNCH TIME AND DATE "
              "(e.g. 0800 EST 01 MAY 1976): Ȩ&dJ_")
  209 FORMAT (5X,"LAUNCH TIME: "4A2/5X"LAUNCH DATE: "6A2)
  210 FORMAT (//"Ȩ&dBLAUNCH VEHICLE (Ȩ&dFSHȨ&dBUTTLE, "
     .        "Ȩ&dFTIȨ&dBTAN, Ȩ&dFDȨ&dBELTA-THOR Ȩ&dF2Ȩ&dB914, "/16X
     .        "Ȩ&dFDȨ&dBELTA-THOR Ȩ&dF3Ȩ&dB914, "
     .        "Ȩ&dFMȨ&dBINUTEȨ&dFMȨ&dBAN II): Ȩ&dJ_")
  211 FORMAT (5X"LAUNCH VEHICLE: "4A2)
  212 FORMAT ("1"80(1H*)/1X,80(1H*)/1X,10(1H*),60X,10(1H*)/
     .        1X,10(1H*)," NASA/MSFC MULTILAYER DIFFUSION MODEL - "
     .        "MOD3A   04 MAY 1977 ",10(1H*)/1X,10(1H*),60X,10(1H*)/
     .        1X,80(1H*)/1X,80(1H*)/
     .        "0RUN "I2" USING DATA FILE "3A2/
     .        "0"3A2,A1" LAUNCH VEHICLE")
  213 FORMAT ("0LAUNCH TIME: "I4" E"A2,4X"DATE: "I2,1XA2,A1,1XI4)
  214 FORMAT ("0PREDICTION TIME: "I4" E"A2,4X"DATE: "I2,1XA2,A1,1XI4/
     .        "0DATA FILE HEADER INFORMATION:")
  215 FORMAT ("0<<<<<OPEN ERROR "I4", PROCESSING CONTINUES WITH "
     .        "NEXT RUN>>>>>")
  216 FORMAT ("0<<<<<READF ERROR "I4", PROCESSING CONTINUES WITH "
     .        "NEXT RUN>>>>>")
  217 FORMAT (6X,40A2)
  218 FORMAT ("0"5X"TIME: "I4" E"A2,4X
     .        "DATE: "I2,1XA2,A1,1XI4)
  219 FORMAT ("1"80(1HS)/1X,20(1HS),40X,20(1HS)/
     .        1X,20(1HS),16X"SOUNDING"16X,20(1HS)/
     .        1X,20(1HS),40X,20(1HS)/1X,80(1HS)//)
  220 FORMAT ("1"80(1HF)/1X,20(1HF),40X,20(1HF)/
     .        1X,20(1HF),16X"FORECAST"16X,20(1HF)/
     .        1X,20(1HF),40X,20(1HF)/1X,80(1HF)//)
  221 FORMAT ("0SURFACE DENSITY (GM/M**3): "F8.2)
  222 FORMAT ("0LAYER   ALTITUDE      DIRECTION  SPEED   TEMP   "
     .        "POT-TEMP  D P TEMP  PRESSURE"/
     .        "  NO. (FEET) (METERS) (DEGREES) (M/SEC)    "
     .        "(DEGREES CENTIGRADE)    (MILLIBARS)")
  223 FORMAT (2XI2,I7,2XI5,7XI3,5XF4.1,4XF4.1,4XF5.2,6XF4.1,6XF5.2)
C
C         TYPE AND DIMENSION STATEMENTS
C
      INTEGER BLANKS,FILE(3),FDIGIT(50),RCHAR,VNAMES(4,5),
              RUNNUM,RA,FO,SDT,TE,ZEROO,GETTD(3),CLDRI(3)
      DIMENSION IPAR(5),VPARS(18,5),IVNAM(5),IDCB(144) IBUF(40),
                IALT(31),DPTEMP(31)
C
C         DATA STATEMENTS
C
C
C*********************************************************************
C***  VPARS(1-18)=SHUTTLE    (19-36)=TITAN    (37-54)=DELTA-THOR 2914
C***        (55-72)=DELTA-THOR 3914    (73-90)=MINUTEMAN II
```

```
C***********************************************************************
C
      DATA VPARS/1.521923E7,6.882968E6,3.441484E6,1.894794173E9,
                 8.56929516E8,1.713859032E9,.6522129891,.4680846,
     .           .375,1479.7,1062.35,1000.0,.1970,.2234,.0316,.2791,
     .           .0002,.64,
     .           5.437528E6,2.718764E6,1.359382E6,3.262516BE8,
     .           1.6312584E8,3.262516BE8,.429580469,.5184223,
     .           5.0,2021.1,1010.55,1000.0,.1932,.2665,.0222,
     .           .2819,.0002,.64,
     .           8.360685E5,9.09811E4,2.729434E5,2.887598E7,
     .           3.14229E6,1.885373E7,.922156,.432703,.54,1766.0,
     .           1000.0,690.0,.1866,.2055,.0156,.3391,.0002,
     .           .50,
     .           1.057557E6,1.482923E5,3.70731E5,6.70269E7,
     .           9.398616E6,4.699308E7,1.245756,.4180947,
     .           0.0,1449.9,1000.0,411.18,.1866,.2055,.0156,.3391,
     .           .0002,.50,
     .           4.684476E5,4.684476E5,1.171119E5,2.8106856E7,
     .           2.8106856E7,2.8106856E7,.469982,.463333,0.0,
     .           2055.9,2055.9,1000.0,.1866,.2055,.0156,.3391,
     .           .0002,.64/
C
      DATA BLANKS/2H  /, RCHAR/1HR/, RA/2HRA/, FO/2HFO/,
     .     SDT/2HDT/, TE/2HTE/, ZEROO/2H00/, NINE9/2H99/,
     .     GETTD/2HGE,2HTT,1HD/, CLDRI/2HCL,2HDR,1HI/
      DATA FDIGIT/2H01,2H02,2H03,2H04,2H05,2H06,2H07,2H08,2H09,2H10,
     .            2H11,2H12,2H13,2H14,2H15,2H16,2H17,2H18,2H19,2H20,
     .            2H21,2H22,2H23,2H24,2H25,2H26,2H27,2H28,2H29,2H30,
     .            2H31,2H32,2H33,2H34,2H35,2H36,2H37,2H38,2H39,2H40,
     .            2H41,2H42,2H43,2H44,2H45,2H46,2H47,2H48,2H49,2H50/
      DATA IVNAM/2HSH,2HTI,2HD2,2HD3,2HMM/
      DATA VNAMES/2HSH,2HUT,2HTL,1HE,
     .            2HTI,2HTA,1HN,1H ,
     .            2HD-,2HT ,2H29,2H14,
     .            2HD-,2HT ,2H39,2H14,
     .            2HMI,2HMM,2HN ,2HII/
C
C         FIND THE LOGICAL UNIT NUMBER OF THE DEVICE TO BE USED FOR
C         INPUT AND SET THE VARIABLE LU EQUAL TO IT
C
      CALL RMPAR(IPAR)
      LU = IPAR(1)
C
C         INITIALIZE SOME COMMON VARIABLES
C
      LTIME = .FALSE.
      YES = 1HY
      PI = 3.141593
      PIOVR2 = 0.5 * PI
      PI43 = 1.3333333 * PI
      TWOPI = 2.0 * PI
      SQR2PI = SQRT(TWOPI)
      DEGRAD = PI/180.0
      RADDEG = 180.0/PI
      DO 2 I=1,3
    2 IOPT(I) = 0
      ZB = 0.0
      ZZ = 0.0
      REFLEC = 1.0
C
C         WRITE THE HEADER OF THE CONSOLE
C
      WRITE (LU,200)
C
C         READ IN THE NUMBER OF RUNS TO BE MADE AND THE FIRST FOUR
C         CHARACTERS OF THE DATA FILE NAMES FOR THOSE RUNS
C
      WRITE (LU,201)
      READ (LU,100) NUMRUN,FILE(1),FILE(2),IFOFF
      NUMRUN = MIN0(MAX0(NUMRUN,1),50)
      IF(IFOFF .GT. 0)IFOFF = IFOFF - 1
      IF(FILE(1) .NE. BLANKS)GO TO 5
      FILE(1) = 2HDA
      FILE(2) = 2HTA
      IFOFF = 0
    5 IF(NUMRUN+IFOFF .GT. 50)NUMRUN = 50 - IFOFF
      DO 6 I=1,NUMRUN
      J = I + IFOFF
    6 WRITE (LU,202) I,FILE(1),FILE(2),FDIGIT(J)
C
C         FIND OUT IF THESE RUNS ARE TO BE RESEARCH RUNS (INTERACTION
C         AND PLOTTING ALLOWED) OR PRODUCTION RUNS
C
      WRITE (LU,203)
      READ (LU,101) I
      IF(I .EQ. RCHAR)IOPT(2) = 1
      IF(IOPT(2) .EQ. 0)GO TO 7
      WRITE (LU,204)
      GO TO 12
    7 WRITE (LU,205)
C
C         FOR PRODUCTION RUNS, READ IN THE TOP OF THE SURFACE LAYER
C         AND THE SIGMA OF THE WIND AZIMUTH ANGLE TO BE USED FOR ALL RUNS
C
      WRITE (LU,206)
      READ (LU,*) TOPSUR
      WRITE (LU,207)
      READ (LU,*) SIGA
C
```

```
C        READ IN AND WRITE OUT THE LAUNCH TIME AND DATE -- IF NOT
C        ENTERED, DO NOT WRITE ANYTHING OUT
C
   12 WRITE (LU,208)
      READ (LU,101) (LAUNTD(I),I=1,10)
      IF(LAUNTD(1) .EQ. BLANKS)GO TO 17
      LTIME = .TRUE.
      CALL CODE
      READ (LAUNTD,102) LTIM,LDAY,LMON,LYEAR
      WRITE (LU,209) (LAUNTD(I),I=1,10)
      GO TO 21
   17 LAUNTD(4) = SDT
C
C        READ IN THE LAUNCH VEHICLE, LET IT DEFAULT IF NOT ENTERED,
C        WRITE IT BACK OUT, AND FILL THE VPAR ARRAY WITH THE
C        APPROPRIATE VEHICLE PARAMETERS
C
   21 WRITE (LU,210)
      READ (LU,101) J
      DO 24 I=1,5
      IF(J .EQ. IVNAM(I))GO TO 25
   24 CONTINUE
      I = 1
   25 IOPT(3) = I - 1
      WRITE (LU,211) (VNAMES(J,I),J=1,4)
      DO 28 J=1,18
   28 VPAR(J) = VPARS(J,I)
C
C        DO LOOP ON THE RUN NUMBER
C
      DO 79 RUNNUM=1,NUMRUN
C
C        SET UP THE FILE NAME FOR THIS RUN, GET THE CURRENT TIME,
C        AND WRITE OUT THE HEADER
C
      FILE(3) = FDIGIT(RUNNUM+IFOFF)
      ASSIGN 31 TO IRETRN
      CALL EXEC(8,GETTD)
   31 CONTINUE
      ITIME = ITIME + 100
      I = IOPT(3) + 1
      WRITE (6,212) RUNNUM,(FILE(J),J=1,3),(VNAMES(J,I),J=1,4)
      IF(LTIME)WRITE (6,213) LTIM,LAUNTD(4),LDAY,LMON(1),LMON(2),LYEAR
      WRITE (6,214) ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
C
C        OPEN THE DATA FILE FOR THIS RUN
C
      CALL OPEN(IDCB,IERR,FILE)
      IF(IERR .GE. 0)GO TO 32
      WRITE (6,215) IERR
      GO TO 79
C
C        READ THE HEADINGS FROM THE DATA FILE, SETTING UP THE
C        APPROPRIATE PARAMETERS
C
   32 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .GE. 0)GO TO 37
   34 WRITE (6,216) IERR
      GO TO 79
   37 IF(IBUF(1) .NE. TE)GO TO 32
   39 WRITE (6,217) (IBUF(I),I=1,LEN)
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 34
      IF(IBUF(1).NE.RA .AND. IBUF(1).NE.FO)GO TO 39
      IOPT(1) = 0
      IF(IBUF(1) .EQ. FO)IOPT(1) = 1
      WRITE (6,217) (IBUF(I),I=1,LEN)
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 34
      WRITE (6,217) (IBUF(I),I=1,LEN)
C
C        READ THE SOUNDING/FORECAST TIME
C
      CALL READF(IDCB,IERR,IBUF,9)
      IF(IERR .LT. 0)GO TO 34
      CALL CODE
      READ (IBUF,103) ISTIM,ISDAY,ISMON(1),ISMON(2),ISYEAR
C
C        CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME
C
      ISTIM = ISTIM - 500
      IF(LAUNTD(4) .NE. 2HST)ISTIM = ISTIM + 100
      IF(ISTIM .GT. 0)GO TO 41
      ISTIM = 2400 + ISTIM
      ISDAY = ISDAY - 1
C
C        WRITE OUT THE NEXT LINE OF THE HEADER
C
   41 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 34
      WRITE (6,217) (IBUF(I),I=1,LEN)
C
C        WRITE OUT THE SOUNDING/FORECAST TIME
C
      WRITE (6,218) ISTIM,LAUNTD(4),ISDAY,ISMON(1),ISMON(2),ISYEAR
C
C        FIND THE FIRST DATA POINT WITH AN ALTITUDE OF 10 FEET
C        OR ABOVE
C
   44 CALL READF(IDCB,IERR,IBUF,40,LEN)
```

```
      IF(IERR .LT. 0)GO TO 34
      CALL B2Z(IBUF(1),J)
      IF(J.LT.ZERO0 .OR. J.GT.NINE9)GO TO 44
      CALL CODE
      READ (IBUF,104) IALT(1),IDIR(1),SPEED(1),TEMP(1),DPTEMP(1),
                     PRESS(1),SURDEN
      IF(IALT(1) .LT. 10)GO TO 44
C
C        TRY TO FIND A TOTAL OF 30 DATA POINTS WITH ALTITUDES
C        BETWEEN 20 FT AND 10,000 FT INCLUSIVE
C
      NUM = 1
      DO 47 I=2,30
   46 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0 .AND. IERR.NE.-12)GO TO 34
      IF(LEN .EQ. -1)GO TO 48
      CALL B2Z(IBUF(1),J)
      IF(J.LT.ZERO0 .OR. J.GT.NINE9)GO TO 46
      CALL CODE
      READ (IBUF,104) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),
                      PRESS(I)
      IF(IALT(I).LT.20 .OR. IALT(I).GT.10000)GO TO 46
   47 NUM = I
C
C        ZERO OUT THE REMAINING ELEMENTS OF THE ARRAYS
C
   48 NUM1 = NUM + 1
      IF(NUM1 .GT. 30)GO TO 51
      DO 49 I=NUM1,30
      ALT(I) = 0.0
      IDIR(I) = 0
      SPEED(I) = 0.0
      TEMP(I) = 0.0
      DPTEMP(I) = 0.0
   49 PRESS(I) = 0.0
C
C        CONVERT TO METRIC UNITS
C
   51 DO 52 I=1,NUM
      ALT(I) = 0.3048 * FLOAT(IALT(I))
   52 SPEED(I) = 0.515 * SPEED(I)
C
C        SORT ALL THE DATA POINTS SO THEY APPEAR IN ASCENDING
C        ORDER OF ALTITUDE
C
      NUM1 = NUM - 1
      DO 58 I=1,NUM1
      JJ = NUM - I
      DO 57 J=1,JJ
      J1 = J + 1
      IF(ALT(J) .LE. ALT(J1))GO TO 57
      ARG = ALT(J)
      ALT(J) = ALT(J1)
      ALT(J1) = ARG
      IARG = IDIR(J)
      IDIR(J) = IDIR(J1)
      IDIR(J1) = IARG
      ARG = SPEED(J)
      SPEED(J) = SPEED(J1)
      SPEED(J1) = ARG
      ARG = TEMP(J)
      TEMP(J) = TEMP(J1)
      TEMP(J1) = ARG
      ARG = DPTEMP(J)
      DPTEMP(J) = DPTEMP(J1)
      DPTEMP(J1) = ARG
      ARG = PRESS(J)
      PRESS(J) = PRESS(J1)
      PRESS(J1) = ARG
   57 CONTINUE
   58 CONTINUE
C
C         CALCULATE THE POTENTIAL TEMPERATURE
C
      DO 62 I=1,NUM
   62 PTEMP(I) = (TEMP(I) + 273.15) * ((1000.0/PRESS(I))**0.288)
C
C         WRITE THE HEADER FOR SOUNDING OR FORECAST
C
      IF(IOPT(1) .EQ. 1)GO TO 64
      WRITE (6,219)
      GO TO 65
   64 WRITE (6,220)
C
C         WRITE THE SURFACE DENSITY AND ALL THE DATA POINTS
C
   65 WRITE (6,221) SURDEN
      WRITE (6,222)
      DO 68 I=1,NUM
      IALTF = 3.281 * ALT(I) + 0.5
      IALTM = ALT(I) + 0.5
      APTEMP = PTEMP(I) - 273.15
   68 WRITE (6,223) I,IALTF,IALTM,IDIR(I),SPEED(I),TEMP(I),
                   APTEMP,DPTEMP(I),PRESS(I)
C
C         TRANSFER TO THE SEGMENT CLDRI -- THE CLOUD RISE MODEL
C
      ASSIGN 75 TO IRETRN
      CALL EXEC(8,CLDRI)
   75 CONTINUE
```

```
C
C         CLOSE THE DATA FILE
C
      CALL CLOSE(IDCB)
C
C         PROCESS THE NEXT RUN
C
   79 CONTINUE
C
C         STOP EXECUTION
C
      STOP
C
C         END OF MOD3A
C
      END
      SUBROUTINE DFEXP(J,CONC)
C
C*************************************************************************
C
C         THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS
C
C         J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER
C         CONC - CONCENTRATION TO BE TESTED
C
C*************************************************************************
C
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,QSPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C         CALCULATE SIGMA Z
C
```

```
      SIGZ = DIST * SIGAP + SIGZ0/1.28
      R3 = 2.0 * SIGZ * SIGZ
C
C        CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION
C
      TWOI = 2.0
      ZT = ALT(J)
      TEMP2 = CLDHT - ZZ
      TEMP3 = CLDHT - 2.0 * ZB + ZZ
      E1 = EXP( - TEMP2 * TEMP2/R3) +
              EXP( - TEMP3 * TEMP3/R3)
    4 TEMP1 = TWOI * (ZT - ZB)
      TEXPSM = E1
      TEXP = (TEMP1 - TEMP2)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP2)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 - TEMP3)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP3)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      IF(E1 .EQ. TEXPSM)GO TO 7
      TWOI = TWOI + 2.0
      GO TO 4
    7 E1 = REFLEC * E1
C
C        CALCULATE SIGMA Y
C
      S8 = DIST * SIGAP + SIGX0
      R2 = SQRT(S8 * S8 + (0.0040589 * FLOAT(IDIR(J) - IDIR(1)) *
                                DIST)**2)
C
C        CALCULATE CLOUD LENGTH
C
      TEMP1 = SPEED(J) - SPEED(1)
      AL1 = 0.28 * TEMP1 * DIST/ASPD
      IF(TEMP1 .GE. 0.0)GO TO 11
      IF(PTEMP(J)-PTEMP(1) .GT. 0.0)AL1 = 0.0
      AL1 = ABS(AL1)
C
C        CALCULATE SIGMA X
C
   11 SIGX = SQRT((AL1/4.3)**2 + SIGX0 * SIGX0)
C
C        IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN
C        TO THE CALLING PROGRAM
C
      IF(CONC .EQ. 1000.0)RETURN
C
C        CALCULATE CROSS WIND DISTANCE
```

```
C
      Y1 = - 2.0 * R2 * R2 * ALOG(15.7496 * CONC * SIGX * R2 *
                                         SIGZ/(Q1 * E1))
      Y1 = SQRT(AMAX1(Y1,0.0))
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF DFEXP
C
      END
      SUBROUTINE ORGIN(IXO,IYO)
C
C*********************************************************************
C
C        THIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING
C        FOR THE COMPLEX AND MAP SELECTED
C
C*********************************************************************
C
C
C        COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEC,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C        INPUT FORMAT STATEMENT
C
  100 FORMAT (I2,1XA1)
C
C        OUTPUT FORMAT STATEMENT
C
  200 FORMAT (//"Ɛ&dBENTER COMPLEX, Ɛ&dFSƐ&dBEA OR Ɛ&dFLƐ&dBAND MAP "
     .        "(e.g. 17,L): Ɛ&dJ_")
```

```
C
C        TYPE AND DIMENSION STATEMENTS
C
      LOGICAL NOT1ST
      INTEGER SCHAR
      DIMENSION IX(8),IY(8)
C
C        DATA STATEMENTS
C
      DATA NOT1ST/.FALSE./, SCHAR/1HS/
      DATA IX/8730,4100,5411,4825,8750,4100,5450,4830/,
           IY/8600,7300,8243,8050,2990,1700,2630,2465/
C
C        IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
C        IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
C        COORDINATES, I, AGAIN
C
      IF(NOT1ST)GO TO 7
C
C        THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
C        AND THE DESIRED MAP, i.e. SEA OR LAND
C
      NOT1ST = .TRUE.
      WRITE (LU,200)
      READ (LU,100) I,J
C
C        CALUCLATE I AS THE INDEX OF THE COORDINATES FOR THE COMPLEX
C        AND MAP ASKED FOR -- DEFAULT IS COMPLEX 17, LAND MAP
C
      K = 0
      IF(J .EQ. SCHAR)K = 4
      J = I - 37
      IF(J.LT.2 .OR. J.GT.4)J = 1
      I = J + K
C
C        SET THE COORDINATES BASED ON THE INDEX I
C
    7 IX0 = IX(I)
      IY0 = IY(I)
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF ORGIN
C
      END
      SUBROUTINE SYMBL(IWIDE,IHI,ISYMB)
      IX=-IWIDE/2
      IY=-IHI/2
      WRITE(12) -1,-1,IX,IY
      WRITE(12,100) IWIDE,0,0,IHI,ISYMB
100   FORMAT(4I5,A1,1H_)
      IY=-IY
      WRITE(12)-1,-1,IX,IY
      RETURN
      END
      SUBROUTINE B2Z(IA,IB)
      IB = IAND(IA,177400B)
      IF(IB .EQ. 020000B)IB = 030000B
      IC = IAND(IA,000377B)
      IF(IC .EQ. 000040B)IC = 000060B
      IB = IOR(IB,IC)
      RETURN
      END
      PROGRAM GETTD,5
C
C**********************************************************************
C
C        THIS SEGMENT RETURNS THE CURRENT TIME, DAY, MONTH, AND YEAR
C
C**********************************************************************
C
C
C        COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATONC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C        TYPE AND DIMENSION STATEMENTS
C
      INTEGER DAYMON(12)
      DIMENSION MONTHS(2,12),IT(5)
C
C        DATA STATEMENTS
```

```
C
      DATA MONTHS/2HJA,1HN,2HFE,1HB,2HMA,1HR,2HAP,1HR,
     .            2HMA,1HY,2HJU,1HN,2HJU,1HL,2HAU,1HG,
     .            2HSE,1HP,2HOC,1HT,2HNO,1HV,2HDE,1HC/
      DATA DAYMON/31,28,31,30,31,30,31,31,30,31,30,31/
C
C        CALL EXEC TO RETURN CURRENT TIME, JULIAN DAY, AND YEAR
C
      CALL EXEC(11,IT,IYEAR)
C
C        USE JUST HOURS AND MINUTES FOR THE TIME
C
      ITIME = 100 * IT(4) + IT(3)
C
C        MAKE APPROPRIATE ADJUSTMENTS IF THIS IS A LEAP YEAR
C
      DAYMON(2) = 28
      I = IYEAR/4
      IF(4*I .EQ. IYEAR)DAYMON(2) = 29
C
C        CONVERT THE JULIAN DAY INTO A MONTH AND A DAY
C
      IDAY = IT(5)
      DO 7 I=1,12
      IDAY = IDAY - DAYMON(I)
      IF(IDAY .LE. 0)GO TO 12
    7 CONTINUE
   12 IDAY = IDAY + DAYMON(I)
      MONTH(1) = MONTHS(1,I)
      MONTH(2) = MONTHS(2,I)
C
C        RETURN TO THE APPROPRIATE PLACE IN MOD3A
C
      GO TO IRETRN
   17 CALL MOD3A
C
C        END OF GETTD
C
      END
      PROGRAM CLDRI,5
C
C***********************************************************************
C
C        CLOUD RISE PROGRAM -- A SEGMENT OF THE MOD3A PROGRAM
C
C***********************************************************************
C
C
C        COMMON BLOCK
C
```

```
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C         OUTPUT FORMAT STATEMENTS
C
  200 FORMAT ("1"27X"EXHAUST CLOUD"/"0LEVEL"4X"ALTITUDE"17X
     .        "RISE TIME"5X"RANGE"6X"DIRECTION"/10X"(METERS)"17X
     .        "(SECONDS)"4X"(METERS)"4X"(DEGREES)")
  201 FORMAT (2XI3,5XF7.1,5X"ADIABATIC"5XF6.1,6XF7.1,7XF5.1)
  202 FORMAT (2XI3,5XF7.1,6X"STABLE"7XF6.1,6XF7.1,7XF5.1)
  203 FORMAT (//"0****CLOUD STABLIZATION****"/
     .        6X"HEIGHT(M): "F6.1/
     .        6X"STABILIZATION TIME AFTER LAUNCH(SEC): "F5.1/
     .        6X"RANGE FROM PAD(M): "F7.1/
     .        6X"DIRECTION FROM PAD(DEG): "F5.1)
  204 FORMAT (//"ESTIMATED TOP OF SURFACE LAYER(M): "F6.1)
  205 FORMAT ("Ȩ&dBDESIRED TOP OF SURFACE LAYER(M): Ȩ&dJ_")
  206 FORMAT (//"0****TOP OF SURFACE LAYER METEOROLOGICAL PARAMETERS"
     .        "****"/
     .        6X"HEIGHT(M): "F6.1/
     .        6X"WIND DIRECTION(DEG): "I3/
     .        6X"WIND SPEED(M/SEC): "F4.1)
  207 FORMAT (//"0****DIFFUSION PARAMETERS****"/
     .        6X"MEAN SPEED(M/SEC): "F4.1/
     .        6X"MEAN TRANSPORT DIRECTION(DEG): "F5.1)
  208 FORMAT (//"Ȩ&dBSIGMA OF WIND AZIMUTH ANGLE, SIGA: Ȩ&dJ_")
  209 FORMAT (//"0SIGMA OF WIND AZIMUTH ANGLE, SIGA: "F4.1)
  210 FORMAT (//"0EFFECTIVE CLOUD HEIGHT(M): "F6.1)
C
C         TYPE AND DIMENSION STATEMENTS
C
      INTEGER CONC(3)
      DIMENSION IAS(31)
C
C         DATA STATEMENT
```

```
C
      DATA CONC/2HCO,2HNC,2H
C
C        INITIALIZE SOME LOCAL VARIABLES
C
C        CRTIME( ) - CLOUD RISE TIME
C        IAS( ) - 0 = ADIABATIC
C                 1 = STABLE
C        ALTINC - ALTITUDE INCREMENT
C        ITERAT - ITERATION COUNTER
C
      RNGY = 0.0
      RNGX = 0.0
      CRTIME(1) = 0.0
      ALTINC = 0.0
      SAVER(1) = 0.0
      SAVEA(1) = 0.0
      ITERAT = 0
C
C        WRITE OUT THE EXHAUST CLOUD HEADER
C
      WRITE (6,200)
C
C        CALCULATE SOME QUANTITIES TO BE USED IN SUBSEQUENT DO LOOP
C
      ALPHAC = 5.12913086E-2 * (TEMP(1) + 273.15) * SURDEN *
     1         GAMMAX**3/(HEATM * QC1)
      GT = 9.8/(TEMP(1) + 273.15)
C
C        DO LOOP TO CALCULATE EXHAUST CLOUD PARAMETERS
C
      DO 9 I=2,NUM
C
      IM1 = I - 1
      IAS(I) = 1
C
C        CALCULATE SLOPE OF POTENTIAL TEMPERATURE, SPEED, AND
C        DIRECTION IN LAYER
C
      DALT = ALT(I) - ALT(IM1)
      GPTEMP = (PTEMP(I) - PTEMP(IM1))/DALT
      GSPEED = (SPEED(I) - SPEED(IM1))/DALT
      GDIR = FLOAT(IDIR(I) - IDIR(IM1))/DALT
C
C        CALCULATE METEOROLOGICAL AND ENERGY FACTOR
C
    2 Z = ALT(I) - ALT(1) - ALTINC
      ALPHA = ALPHAC * Z**4/(AA * Z**BB + CC)
C
C        CALCULATE POTENTIAL TEMPERATURE FACTOR
```

```
C
      STAB = GT * (PTEMP(I) - ALTINC * GPTEMP - PTEMP(1))/
                 (ALT(I) - ALTINC - ALT(1) + 1.0E-7)
C
C        CALCULATION FOR ADIABATIC RISE
C
      IF(STAB .GT. 0.000001)GO TO 4
      CRTIME(I) = SQRT(ALPHA)
      IAS(I) = 0
      GO TO 6
C
C        CALCULATION FOR STABLE CLOUD RISE
C
C        C2 - ARGUMENT OF ARC COSINE (MUST BE LESS THAN -1)
C
    4 C2 = 1.0 - 0.5 * ALPHA * STAB
      IF(C2 .LT. -1.0)GO TO 5
      C3 = C2/SQRT(1.0 - C2 * C2)
      CRTIME(I+ITERAT) = (PIOVR2 - ATAN(C3))/SQRT(STAB)
      IF(ITERAT .EQ. 1)GO TO 11
      GO TO 6
C
C        ITERATE IN LAYER
C
    5 ALTINC = ALTINC + 5.0
      ITERAT = 1
      GO TO 2
C
C        CALCULATE RANGE AND DIRECTION
C
    6 DELRNG = - 0.5 * (SPEED(IM1) + SPEED(I)) *
                     (CRTIME(IM1) - CRTIME(I))
      DELDIR = 0.00872665 * FLOAT(IDIR(I) + IDIR(IM1))
      RNGY = RNGY - DELRNG * SIN(DELDIR)
      RNGX = RNGX - DELRNG * COS(DELDIR)
      AZMUTH = RADDEG * ATAN2(RNGY,RNGX)
      IF(AZMUTH .LT. 0.0)AZMUTH = AZMUTH + 360.0
      DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
      SAVER(I) = DELRNG
      SAVEA(I) = AZMUTH
C
C        WRITE OUT THE VARIABLES WITH THE APPROPRIATE FORMAT STATEMENT
C        BASED OF WHETHER OR NOT CLOUD IS ADIABATIC OR STABLE
C
      IF(IAS(I) .NE. 0)GO TO 8
      WRITE (6,201) I,ALT(I),CRTIME(I),DELRNG,AZMUTH
      GO TO 9
    8 WRITE (6,202) I,ALT(I),CRTIME(I),DELRNG,AZMUTH
    9 CONTINUE
C
```

```
C         CALCULATE AND WRITE OUT STABILIZATION HEIGHT AND TIME
C
   11 DELRNG = 0.5 * (SPEED(IM1) - ALTINC * GSPEED + SPEED(I)) *
     +                  (CRTIME(I + 1) - CRTIME(IM1))
      DALT = 0.00872665 * (FLOAT(IDIR(I) + IDIR(IM1)) - GDIR * ALTINC)
      RNGY = RNGY - DELRNG * SIN(DALT)
      RNGX = RNGX - DELRNG * COS(DALT)
      AZMUTH = RADDEG * ATAN2(RNGY,RNGX)
      IF(AZMUTH .LT. 0.0)AZMUTH = AZMUTH + 360.0
      DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
      ALT(31) = ALT(I) - ALTINC
      WRITE (6,203) ALT(31),CRTIME(I+1),DELRNG,AZMUTH
C
C         STORE THE INDEX OF THE ESTIMATED TOP OF THE SURFACE LAYER
C
      JTOP = I + 1
C
C         LOAD THE CLOUD RISE TIME ARRAY
C
      CRTIME(31) = CRTIME(JTOP)
      DO 15 J=I,NUM
   15 CRTIME(I) = CRTIME(31)
C
C         IS THIS A RESEARCH OR A PRODUCTION RUN?
C
      IF(IOPT(2) .NE. 0)GO TO 22
C
C         PRODUCTION RUN -- IF TOPSUR IS UNDEFINED, USE JTOP AS ESTIMATED
C
   17 IF(TOPSUR .LE. 0.0)GO TO 24
C
C         CALCULATE JTOP BASED ON VALUE OF TOPSUR
C
      LEASTD = 9999999.9
      DO 19 I=1,NUM
      DIFF = ABS(ALT(I) - TOPSUR)
      IF(DIFF .GT. LEASTD)GO TO 19
      LEASTD = DIFF
      JTOP = I
   19 CONTINUE
      GO TO 24
C
C         WRITE OUT THE ESTIMATED TOP OF SURFACE LAYER -- READ IN
C         THE ONE TO BE USED -- CALCULATE JTOP
C
   22 WRITE (LU,204) ALT(JTOP)
      WRITE (LU,205)
      READ (LU,*) TOPSUR
      GO TO 17
C
```

```
C         WRITE OUT THE TOP OF THE SURFACE LAYER AND WIND DIRECTION
C         AND SPEED AT THE TOP
C
   24 TOPSUR = ALT(JTOP)
      WRITE (6,206) TOPSUR,IDIR(JTOP),SPEED(JTOP)
C
C         CALCULATE SOURCE STRENGTH
C
      Q1 = 1.289E9 * (TEMP(1) + 273.15)/PRESS(1) * TOPSUR**0.4037
C
C         CALCULATE AND WRITE OUT THE MEAN WIND SPEED, ASPD, AND
C         DIRECTION, ADIR
C
      DO 28 I=2,JTOP
      IF(IABS(IDIR(I) - IDIR(I - 1)) .LT. 180)GO TO 28
      DO 27 J=1,JTOP
   27 IF(IDIR(J) .LT. 180)IDIR(J) = IDIR(J) + 360
      GO TO 31
   28 CONTINUE
C
   31 ASPD = 0.0
      ADIR = 0.0
      DO 32 I=2,JTOP
      IM1 = I - 1
      DALT = ALT(I) - ALT(IM1)
      ASPD = ASPD + 0.5 * (SPEED(I) + SPEED(IM1)) * DALT
   32 ADIR = ADIR + 0.5 * FLOAT(IDIR(I) + IDIR(IM1)) * DALT
C
      DO 34 I=1,JTOP
   34 IF(IDIR(I) .GT. 360)IDIR(I) = IDIR(I) - 360
C
      DALT = ALT(JTOP) - ALT(1)
      ASPD = ASPD/DALT
      ADIR = ADIR/DALT
      IF(ADIR .GT. 180.0)GO TO 35
      ADIR = ADIR + 180.0
      GO TO 36
   35 ADIR = ADIR - 180.0
C
   36 WRITE (6,207) ASPD,ADIR
C
C         IS THIS A RESEARCH OR A PRODUCTION RUN?
C
      IF(IOPT(2) .EQ. 0)GO TO 45
C
C         RESEARCH RUN -- READ IN SIGA
C
      WRITE (LU,208)
      READ (LU,*) SIGA
C
```

```
C         WRITE OUT SIGA, THE SIGMA OF THE WIND AZIMUTH ANGLE
C
   45 WRITE (6,209) SIGA
C
      SIGAP = 0.0087266 * SIGA
C
C         CALCULATE THE HORIZONTAL AND VERTICAL CLOUD DIMENSIONS,
C         i.e. SIGX0 AND GSPEED
C
      SIGX0 = 0.297674 * ALT(31)
      GSPEED = 0.232558 * ALT(31)
C
C         CALCULATE AND WRITE OUT THE EFFECTIVE CLOUD HEIGHT, CLDHT
C
      CLDHT = ALT(31)
      CLDRAD = 2.15 * SIGX0
      IV2 = 0
      IF(CLDRAD+ALT(31) .GE. ALT(JTOP))IV2 = 1
      SIGZ0 = SIGX0
      IF(IV2 .EQ. 1)SIGZ0 = (ALT(JTOP) - ALT(31) + CLDRAD)/4.3
      IF(SIGZ0 .GT. 0.0)GO TO 47
      CLDHT = 0.5 * ALT(JTOP)
      SIGZ0 = 0.64 * CLDHT/2.15
      GO TO 49
   47 IF(IV2 .EQ. 1)CLDHT = 0.5 * (ALT(JTOP) + ALT(31) - CLDRAD)
C
   49 WRITE (6,210) CLDHT
C
C         CALL THE SEGMENT CONC
C
      CALL EXEC(8,CONC)
C
C         END OF CLDRI
C
      END
      PROGRAM CONC,5
C
C**********************************************************************
C
C         CONCENTRATION AND DOSAGE PROGRAM -- A SEGMENT OF THE
C         MOD3A PROGRAM
C
C**********************************************************************
C
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .          LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .          Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .          SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
     .          TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,YI,NUMRUN,
     .          YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C         INPUT FORMAT STATEMENT
C
  100 FORMAT (A1)
C
C         OUTPUT FORMAT STATEMENTS
C
  201 FORMAT (//"Ɛ&dBCENTERLINE CONCENTRATION PLOT DESIRED? "
     .           "(Ɛ&dFYƐ&dBES OR Ɛ&dFNƐ&dBO): Ɛ&dJ_")
  202 FORMAT (5X"NO")
  203 FORMAT (5X"YES")
  204 FORMAT ("1"12X"CLOUD CONCENTRATIONS AND DOSAGES"/
     .          "0DISTANCE"4X"CONCENTRATION"5X"DOSAGE"6X
     .           "TIME AFTER LAUNCH(SEC)"/
     .           " (METERS)"8X"(PPM)"8X"(PPM SEC)"8X"START"3X"FINISH")
  205 FORMAT (1XF7.1,8XF7.3,8XF7.3,9XF5.1,3XF5.1)
  206 FORMAT (//"0****POINT OF MAXIMUM CONCENTRATION****"/
     .           6X"RANGE FROM PAD(M): "F8.1/
     .           6X"DIRECTION(DEG): "F5.1/
     .           6X"HEIGHT(M): 2.0"/
     .           6X"MAXIMUM CONCENTRATION(PPM): "F6.3)
  207 FORMAT (//"Ɛ&dBOFF-CENTER CONCENTRATIONS DESIRED? "
     .           "(Ɛ&dFYƐ&dBES OR Ɛ&dFNƐ&dBO): Ɛ&dJ_")
  208 FORMAT (//"0****CONCENTRATIONS AND DOSAGES WITH 10 DEGREE "
     .           "UNCERTAINTIES****")
  209 FORMAT (/"Ɛ&dBRANGE(M), AZIMUTH(DEG) "
     .           "(0 RANGE TERMINATES PROCEDURE): Ɛ&dJ_")
  210 FORMAT ("0"5X"RANGE(M): "F7.1/
     .           6X"AZIMUTH(DEG): "F5.1/
     .           6X"MATERIAL"5X"CONCENTRATION(PPM)"11X"DOSAGE(PPM)")
  211 FORMAT (4I5,I2)
  212 FORMAT (7X3A2,6XF8.3" +/- "F8.3,4XF8.3" +/- "F8.3)
  213 FORMAT (//"Ɛ&dBISOPLETH PLOT DESIRED? "
     .           "(Ɛ&dFYƐ&dBES OR Ɛ&dFNƐ&dBO): Ɛ&dJ_")
C
C         TYPE AND DIMENSION STATEMENTS
```

```
C
      LOGICAL IGRAF
      DIMENSION FACT(3),CMNPL(3),DMNPL(3),MATS(3,5),ISOPO(3)
C
C        DATA STATEMENTS
C
      DATA FACT/0.0,-0.174533,0.174533/
      DATA MATS/2H  ,2HNC,2HL ,2H  ,2H C,2HO ,
     .          2H  ,2HCO,2H2 ,2H A,2HL2,2HO3,
     .          2H  ,2H N,2HO /
      DATA ISOPO/2HIS,2HOP,1HO/
C
C        IF THIS IS A RESEARCH RUN, DETERMINE IF PLOTTING IS DESIRED
C
      IF(IOPT(2) .EQ. 0)GO TO 55
C
      WRITE (LU,201)
      READ (LU,100) I
      IF(I .EQ. YES)GO TO 54
      IGRAF = .FALSE.
      WRITE (LU,202)
      GO TO 55
   54 IGRAF = .TRUE.
      WRITE (LU,203)
C
C        DO LOOP FOR CONCENTRATION AND DOSAGE CALCULATIONS
C
C        DIST - RANGE FROM STABILIZATION
C        DOSPK - DOSAGE
C        DOSMAX - MAXIMUM DOSAGE
C        CONCPK - CONCENTRATION
C        CONMAX - MAXIMUM CONCENTRATION
C
   55 CONMAX = 0.0
      DOSMAX = 0.0
      ACTVOL = PI43 * CLDRAD * CLDRAD * CLDRAD
      TOTVOL = ACTVOL
      IF(IW2 .EQ. 1)ACTVOL = PI * (ALT(JTOP) + CLDRAD - ALT(31))**2 *
     .                       (2.0 * CLDRAD - ALT(JTOP) + ALT(31))/3.0
      Q1 = Q1 * ACTVOL/TOTVOL
C
      WRITE (6,204)
C
      DO 59 I=0,20000,250
C
      DIST = I
C
      CALL DFEXP(JTOP,1000.0)
C
      DOSPK = Q1 * E1/(TWOPI * R2 * ASPD * SQRT(0.5 * R3))
      CONCPK = DOSPK * ASPD/(SQR2PI * SIGX)
C
      IF(IGRAF)CALL CPLOT
C
      DOSMAX = AMAX1(DOSPK,DOSMAX)
C
      IF(CONCPK .LE. CONMAX)GO TO 58
      RATOMC = DIST
      CONMAX = CONCPK
      SGXMAX = SIGX
      SGYMAX = SIGY
C
   58 IF(AMOD(DIST,1000.0) .NE. 0.0)GO TO 59
C
      ARG1 = CRTIME(31) + (DIST - AL1)/ASPD
      ARG2 = CRTIME(31) + (DIST + AL1)/ASPD
      WRITE (6,205) DIST,CONCPK,DOSPK,ARG1,ARG2
C
   59 CONTINUE
C
C        CALCULATE AND WRITE OUT THE POINT OF MAXIMUM CONCENTRATION
C
      ARG1 = DEGRAD * ADIR
      DIST = RATOMC * COS(ARG1)
      Y1 = RATOMC * SIN(ARG1)
C
      DO 62 I=2,JTOP
      IF(CLDHT .LE. ALT(I))GO TO 63
   62 CONTINUE
      I = JTOP
C
   63 IM1 = I - 1
      RANGSR = SAVER(IM1) + (CLDHT - ALT(IM1)) *
                 (SAVER(I) - SAVER(IM1))/(ALT(I) - ALT(IM1))
C
      ARG1 = SAVEA(I) - SAVEA(IM1)
      IF(ABS(ARG1) .LT. 180.0)GO TO 66
      IF(ARG1 .GT. 0.0)GO TO 65
      SAVEA(I) = SAVEA(I) + 360.0
      GO TO 66
   65 SAVEA(IM1) = SAVEA(IM1) + 360.0
C
   66 AZCS = SAVEA(IM1) + (CLDHT - ALT(IM1)) * (SAVEA(I) - SAVEA(IM1))/
                           (ALT(I) - ALT(IM1))
      IF(AZCS .GE. 360.0)AZCS = AZCS - 360.0
C
      ARG1 = DEGRAD * AZCS
      X2 = RANGSR * COS(ARG1)
      Y2 = RANGSR * SIN(ARG1)
      X = DIST + X2
      Y = Y1 + Y2
C
      RNGE = SQRT(X * X + Y * Y)
      DIR = RADDEG * ATAN2(Y,X)
      IF(DIR .LT. 0.0)DIR = DIR + 360.0
      WRITE (6,206) RNGE,DIR,CONMAX
C
C        IF THIS IS A PRODUCTION RUN, SKIP THE OFF CENTER CONCENTATION
C        SECTION AND THE CALL OF SEGMENT ISOPO -- IF PLOTTING WAS NOT
C        REQUESTED, JUST SKIP THE OFF CENTER CONCENTRATION SECTION
C
      IF(IGRAF)GO TO 68
      IF(IOPT(2) .EQ. 0)GO TO 88
      GO TO 81
C
C        OFF CENTER CONCENTRATIONS SECTION
C
   68 CALL LABEL(JTOP)
C
C        ARE OFF CENTER CONCENTRATIONS DESIRED?
C
      WRITE (LU,207)
      READ (LU,100) I
      IF(I .NE. YES)GO TO 78
C
C        OFF CENTER CONCENTRATIONS ARE DESIRED
C
      WRITE (LU,203)
      WRITE (6,208)
C
      CALL ORGIN(IXSET,IYSET)
C
      ARG1 = 0.0
      IF(ADIR .GT. 180.0)ARG1 = 360.0
      BETAF = DEGRAD * (180.0 + ARG1 - ADIR)
C
      ARG1 = 0.0
      IF(AZCS .GT. 180.0)ARG1 = 360.0
      BETAS = DEGRAD * (180.0 + ARG1 - AZCS)
      XP = RANGSR * COS(BETAS)
      YP = RANGSR * SIN(BETAS)
C
      ITER = 0
C
C        LOOP ON OFF CENTER CONCENTRATION REQUESTS
C
   71 ITER = ITER + 1
C
C        READ IN AND WRITE OUT THE RANGE AND AZIMUTH FOR THE
C        OFF CENTER CONCENTRATION CALCULATION -- ENTERING A RANGE OF 0
```

```
C         TERMINATES THE PROCEDURE
C
      WRITE (LU,209)
      READ (LU,*) RP,AZP
      IF(RP .LE. 0.0)GO TO 81
      WRITE (6,210) RP,AZP
C
      ARG1 = 0.0
      IF(AZP .GT. 180.0)ARG1 = 360.0
      AP = DEGRAD * (180.0 + ARG1 - AZP)
      XS = RP * COS(AP)
      YS = RP * SIN(AP)
C
C         ON THE PLOTTER, WRITE OUT AN ASTERISK AND THE ITERATION
C         NUMBER AT THE LOCATION WHERE THE OFF CENTER CONCENTRATION
C         CALCULATION IS DESIRED
C
      IX = IXSET + 0.2631 * XS
      IY = IYSET + 0.3545 * YS
      WRITE (12) -1,1,IX,IY
      CALL SYMBL(100,125,1H*)
      IX = IX + 75
      WRITE (12) -1,1,IX,IY
      WRITE (12,211) 100,0,0,125,ITER
C
C         CALCULATE THE CONCENTRATIONS AND DOSAGES AT THIS POINT PLUS
C         10 DEGREES UNCERTAINTIES ON EITHER SIDE
C
      XHAT = XS - XP
      YHAT = YS - YP
C
      DO 74 I=1,3
      ARG1 = BETAF - FACT(I)
      Y = - XHAT * SIN(ARG1) + YHAT * COS(ARG1)
      CALL DFEXP(JTOP,1000.0)
      DOS = Q1 * E1 * EXP(- Y * Y/(2.0 * R2 * R2))/
     *      (TWOPI * R2 * ASPD * SQRT(0.5 * R3))
      CONC = DOS * ASPD/(SQR2PI * SIGX)
      CMNPL(I) = CONC
   74 DMNPL(I) = DOS
C
C         CALCULATE AND WRITE OUT THE CONCENTRATION AND DOSAGE FOR
C         EACH MATERIAL
C
      DELC = ABS(0.5 * (2.0 * CMNPL(1) - CMNPL(2) - CMNPL(3)))
      DELD = ABS(0.5 * (2.0 * DMNPL(1) - DMNPL(2) - DMNPL(3)))
      WRITE (6,212) (MATS(I,1),I=1,3),CMNPL(1),DELC,DMNPL(1),DELD
C
      ARG1 = PCO/PHCL
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,212) (MATS(I,2),I=1,3),CONC,DLC,DOS,DLD
C
      ARG1 = PCO2/PHCL
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,212) (MATS(I,3),I=1,3),CONC,DLC,DOS,DLD
C
      ARG1 = PAL203/PHCL * 0.43882420 * PRESS(1)/
                           (TEMP(1) + 273.16)
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,212) (MATS(I,4),I=1,3),CONC,DLC,DOS,DLD
C
      ARG1 = PNO/PHCL
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,212) (MATS(I,5),I=1,3),CONC,DLC,DOS,DLD
C
C         REQUEST ANOTHER POINT FOR AN OFF CENTER CONCENTRATION
C         CALCULATION
C
      GO TO 71
C
C         OFF CENTER CONCENTRAIONS ARE NOT DESIRED
C
   78 WRITE (LU,202)
C
C         IS AN ISOPLETH PLOT DESIRED?
C
   81 WRITE (LU,213)
      READ (LU,100) I
C
C         IF AN ISOPLETH PLOT IS DESIRED, CALL THE SEGMENT ISOPO
C
      IF(I .NE. YES)GO TO 87
      WRITE (LU,203)
      CALL EXEC(8,ISOPO)
   87 WRITE (LU,202)
C
C         RETURN TO THE APPROPRIATE PLACE IN MOD3A
C
```

```
   88 GO TO IRETRN
   89 CALL MOD3A
C
C         END OF CONC
C
      END
      SUBROUTINE CPLOT
C
C**********************************************************************
C
C         THIS SUBROUTINE PLOTS THE CONCENTRATION AND DOSAGE CENTERLINE
C
C**********************************************************************
C
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C
C
      IEXPC=0
      IEXPD=IEXPC+2
      IX=DIST*9295./30000.+725.
      IYC=CONCPK*8231./10.**(IEXPC+1)+1040.
      IYD=DOSPK*8231./10.**(IEXPD+1)+1040.
      IF(DIST.NE.0.) GO TO 30
      WRITE(12) -1,1,IX,IYD
      CALL SYMBL(100,100,25400B)
      WRITE(12) -1,1,IX,IYC
C
C         RETURN TO THE CALLING PROGRAM
C
      RETURN
```

```
30    WRITE(12) 1,1,IX,IYC
      WRITE(12) -1,1,IX,IYD
      CALL SYMBL(100,100,25400B)
      WRITE(12) -1,1,IX,IYC
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF CPLOT
C
      END
      SUBROUTINE LABEL(J2)
C
C***********************************************************************
C
C        THIS SUBROUTINE LABELS THE CONCENTRATION AND DOSAGE CENTERLINE
C        PLOTS
C
C***********************************************************************
C
C
C        COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .           (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .           (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .           (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .           (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .           (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C        OUTPUT FORMAT STATEMENTS
C
  200 FORMAT (4I5,I2)
  201 FORMAT (4I5,F5.0)
  202 FORMAT (4I5,F5.2)
  203 FORMAT (4I5,I4" E"A2,2XI2,1XA2,A1,1XI4)
C
C        LABEL THE PLOT
```

```
C
      IEXPC=0
      IEXPD=IEXPC+2
      NEXPC=-IEXPC
      NEXPD=-IEXPD
      WRITE (12) -1,1,300,5000
      WRITE (12,200) 0,150,-100,0,NEXPC
      WRITE (12) -1,1,300,6500
      WRITE (12,200) 0,150,-100,0,NEXPD
      WRITE (12) -1,1,3700,8950
      WRITE (12,201) 125,0,0,125,CLDHT
      WRITE (12) -1,1,3700,8745
      WRITE (12,201) 125,0,0,125,CRTIME(31)
      WRITE (12) -1,1,3700,8540
      WRITE (12,202) 125,0,0,125,CONMAX
      WRITE (12) -1,1,3700,8335
      WRITE (12,201) 125,0,0,125,ALT(J2)
      WRITE (12) -1,1,3700,8130
      WRITE (12,201) 125,0,0,125,0.
      WRITE (12) -1,1,3700,7925
      WRITE (12,201) 125,0,0,125,0.0
      IF(IOPT(1) .EQ. 1)GO TO 4
      WRITE (12) -1,1,5625,8980
      WRITE (12) 1,1,6125,8980
      GO TO 7
    4 WRITE (12) -1,1,5025,8980
      WRITE (12) 1,1,5525,8980
      WRITE (12) -1,1,5725,8950
      WRITE (12,203) 125,0,0,125,ISTIM,LAUNTD(4),ISDAY,ISMON,ISYEAR
    7 WRITE (12) -1,1,5725,8695
      WRITE (12,203) 125,0,0,125,ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
      WRITE (12) -1,1,5725,8490
      IF(LTIME)WRITE (12,203) 125,0,0,125,LTIM,LAUNTD(4),LDAY,LMON,LYEAR
C
C        RETURN TO CONC
C
      RETURN
C
C        END OF LABEL
C
      END
      PROGRAM ISOP0,5
C
C*********************************************************************
C
C         ISOPLETH PLOTTING PROGRAM -- A SEGMENT OF THE MOD3A PROGRAM
C
C*********************************************************************
C
C
```

```
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       Q1,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SICZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLG1(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C         INPUT FORMAT STATEMENT
C
  100 FORMAT (A1)
C
C         OUTPUT FORMAT STATEMENTS
C
  200 FORMAT ("1"20X"CLOUD LOCATION AND DIMENSIONS"/
     .        " TIME FROM CLOUD STABILIZATION"5X"RANGE"5X"AZIMUTH"
     .        8X"DIAMETERS (METERS)"/
     .        11X"(MINUTES)"14X"(METERS)"4X"(DEG)"6X"CROSS WIND"
     .        4X"ALONG WIND")
  201 FORMAT (12XF6.2,16XF8.1,4XF5.1,7XF7.1,7XF7.1)
  202 FORMAT (//"Ɛ&dBDEFAULT ISOPLETH CONCENTRATION VALUES? "
     .        "(Ɛ&dFYƐ&dBES OR Ɛ&dFNƐ&dBO): Ɛ&dJ_")
  203 FORMAT (//"Ɛ&dBISOPLETH CONCENTRATION VALUE "
     .        "(NEGATIVE VALUE TERMINATES PROCEDURE): Ɛ&dJ_")
  204 FORMAT (4I5,I4" E"A2,2XI2,1XA2,A1,1XI4)
  205 FORMAT (4I5,A1)
  206 FORMAT (4I5,F5.2"_")
  207 FORMAT (4I5", "F5.2"_")
C
C         TYPE AND DIMENSION STATEMENTS
C
      LOGICAL DFALTC
      DIMENSION CONC(10)
C
C         DETERMINE THE ORIGIN ON THE MAP FOR THIS PLOT AND MOVE THE
C         PEN THERE
C
      CALL ORGIN(IXO,IYO)
      WRITE (12) -1,1,IXO,IYO
C
C        DETERMINE THE INDEX IN THE ALTITUDE DATA ARRAY THAT HAS
C        THAT ALTITUDE JUST LOWER THAN THE EFFECTIVE CLOUD HEIGHT, CLDHT
C
      DO 4 I=2,JTOP
      IF(CLDHT .GT. ALT(I))GO TO 4
      ICLDHT = I - 1
      GO TO 5
    4 CONTINUE
      ICLDHT = JTOP
C
C        DRAW THE LINE DEPICTING CLOUD MOVEMENT ALONG THE GROUND
C        AS FAR AS THE CLOUD STABILIZATION POINT
C
    5 X = 0.0
      Y = 0.0
      DO 9 I=2,ICLDHT
      IM1 = I - 1
      RANGE = 0.5 * (CRTIME(I) - CRTIME(IM1)) * (SPEED(I) + SPEED(IM1))
      DIR = 0.5 * FLOAT(IDIR(I) + IDIR(IM1))
      IF(IABS(IDIR(I) - IDIR(IM1)) .GT. 180)DIR = DIR - 180.0
      IF(DIR .LT. 0.0)DIR = DIR + 360.0
      DIR = DEGRAD * (360.0 - DIR)
      X = X + RANGE * COS(DIR)
      Y = Y + RANGE * SIN(DIR)
      IX = INT(0.2631 * X) + IXO
      IY = INT(0.3545 * Y) + IYO
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 11
    9 WRITE (12) 1,1,IX,IY
C
C        MAKE THE CALCULATIONS NECESSARY TO WRITE OUT THE CLOUD
C        LOCATION AND DIMENSIONS
C
   11 ALT1 = 0.5 * (CLDHT + ALT(ICLDHT))
      ICLDP1 = ICLDHT + 1
      ARG1 = ALT(ICLDP1) - ALT(ICLDHT)
      ARG2 = (CLDHT - ALT(ICLDHT))/ARG1
      SPCENT = SPEED(ICLDHT) + (SPEED(ICLDP1) - SPEED(ICLDHT)) * ARG2
      RANGE = SPCENT * (CRTIME(ICLDP1) - CRTIME(ICLDHT)) * ARG2
      IF(IABS(IDIR(ICLDP1) - IDIR(ICLDHT)) .LT. 180)GO TO 14
      IF(IDIR(ICLDP1) .LT. 180)IDIR(ICLDHT) = IDIR(ICLDHT) + 360
      IF(IDIR(ICLDHT) .LT. 180)IDIR(ICLDHT) = IDIR(ICLDHT) + 360
   14 DIR = FLOAT(IDIR(ICLDHT)) + (ALT1 - ALT(ICLDHT)) *
     .                          FLOAT(IDIR(ICLDP1) - IDIR(ICLDHT))/ARG1
      IF(DIR .GT. 360.0)DIR = DIR - 360.0
      IF(DIR .GT. 180.0)GO TO 17
      DIR = DIR + 180.0
      GO TO 18
   17 DIR = DIR - 180.0
   18 DIR = 180.0 - DIR
      ARG1 = DEGRAD * DIR
      X = X + RANGE * COS(ARG1)
      Y = Y + RANGE * SIN(ARG1)
      R = SQRT(X * X + Y * Y)
      DELR = 300.0 * ASPD
C
      DACRS = 4.30 * SIGXO
      DALNG = 4.30 * SIGXO
C
      ARG1 = 180.0
      IF(DIR .GT. 180.0)ARG1 = 540.0
      AZ = ARG1 - DIR
C
      ARG1 = 180.0
      IF(ADIR .GT. 180.0)ARG1 = 540.0
      DAZ = ARG1 - ADIR
      ARG1 = DEGRAD * DAZ
      DELX = DELR * COS(ARG1)
      DELY = DELR * SIN(ARG1)
C
      DELU = ABS(SPEED(ICLDHT) - SPEED(1))
C
      DELTH = IDIR(JTOP) - IDIR(1)
C
      TIM = 0.0
      R1 = 0.0
      XC = X
      YC = Y
      TXL = 0.28 * DELU/ASPD
      SIGXO2 = SIGXO * SIGXO
      S82 = S8 * S8
      WRITE (6,200)
C
      DO 22 I=1,13
      WRITE (6,201) TIM,R,AZ,DACRS,DALNG
      TIM = TIM + 5.0
      R1 = R1 + DELR
      XL = R1 * TXL
      SIGX = SQRT((XL/4.30)**2 + SIGXO2)
      DACRS = 4.30 * SIGX
      SIGY = SQRT(S82 + (0.0040589 - 3.0 * DELTH * R1)**2)
      DALNG = 4.30 * SIGY
      XC = XC + DELX
      YC = YC + DELY
      R = SQRT(XC * XC + YC * YC)
   22 AZ = 180.0 - RADDEG * ATAN2(YC,XC)
C
C        LABEL THE CLOUD STABILIZATION POINT WITH A +
C
```

```
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 77
      IXX = IX
      IYY = IY
      WRITE (12) 1,1,IX,IY
      CALL SYMBL(150,150,1H+)
C
C        LABEL THE POINT OF MAXIMUM CONCENTRATION WITH A @
C
      DIR = DEGRAD * (180.0 - ADIR)
      CDIR = COS(DIR)
      SDIR = SIN(DIR)
      IX1 = INT(0.2631 * (X + RATOMC * CDIR)) + IX0
      IY1 = INT(0.3545 * (Y + RATOMC * SDIR)) + IY0
      WRITE (12) -1,1,IX1,IY1
      CALL SYMBL(150,150,1H@)
C
C        DRAW THE LINE OF CLOUD MOVEMENT ALONG THE GROUND FROM
C        THE CLOUD STABILIZATION POINT ON
C
      WRITE (12) -1,1,IXX,IYY
      RANGE = 1000.0
   27 X = X + RANGE * CDIR
      Y = Y + RANGE * SDIR
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 29
      WRITE (12) 1,1,IX,IY
      GO TO 27
   29 WRITE (12) -1,1,IXX,IYY
C
C        ARE DEFAULT CONCENTRATION VALUES GOING TO BE USED
C        FOR THE PLOTS
C
      WRITE (LU,202)
      READ (LU,100) I
      DFALTC = .FALSE.
      IF(I .NE. YES)GO TO 35
C
C        YES -- SET UP THE DEFAULT VALUES
C
      DFALTC = .TRUE.
      CONC(1) = 0.1 * CONMAX
      CONC(2) = 0.5 * CONMAX
      CONC(3) = 0.75 * CONMAX
      CONC(4) = - 1.0
C
C        DO LOOP OVER THE 10 POSSIBLE CONCENTRATION VALUES FOR THE PLOTS
C
```

```
   35 DO 58 I=1,10
C
C           IF DEFAULT CONCENTRATION VALUES ARE NOT BEING USED,
C           READ IN THE VALUE FOR THIS PLOT
C
      IF(DFALTC)GO TO 37
      WRITE (LU,203)
      READ (LU,*) CONC(I)
   37 IF(CONC(I) .LT. 0.0)GO TO 61
C
C           ITERATE TO FIND THE LOCATION OF THIS CONCENTRATION
C           ON THE PLOT
C
      DIST = 0.0
      DINC = 1000.0
C
   41 CALL DFEXP(JTOP,CONC(I))
      IF(Y1 .GT. 0.0)GO TO 42
      DIST = DIST + DINC
      GO TO 41
C
   42 IF(DINC .LE. 100.0)GO TO 43
      DIST = DIST - 900.0
      DINC = 100.0
      GO TO 41
C
   43 IF(DINC .LE. 10.0)GO TO 44
      DIST = DIST - 90.0
      DINC = 10.0
      GO TO 41
C
C           PLOT OUT THE CONCENTRATION LINE ON BOTH SIDES
C
   44 DIST = DIST - 10.0
      IX1 = INT(0.2631 * DIST * CDIR) + IXX
      IY1 = INT(0.3545 * DIST * SDIR) + IYY
      IF(IX1.LT.0 .OR. IX1.GT.9999 .OR. IY1.LT.0 .OR. IY1.GT.9999)
                                                                GO TO 58
      WRITE (12) -1,1,IX1,IY1
C
      DIST = DIST + 10.0
      IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
      IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 58
      WRITE (12) 1,1,IX,IY
C
      WRITE (12) -1,1,IX1,IY1
C
      IX2 = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
      IY2 = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
      IF(IX2.LT.0 .OR. IX2.GT.9999 .OR. IY2.LT.0 .OR. IY2.GT.9999)
                                                                  GO TO 58
C
   46 WRITE (12) 1,1,IX2,IY2
      WRITE (12) -1,1,IX,IY
C
      IX1 = IX2
      IY1 = IY2
      DIST = DIST + 500.0
      CALL DFEXP(JTOP,CONC(I))
      IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
      IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 58
      WRITE (12) 1,1,IX,IY
C
      IF(Y1 .GT. 0.0)GO TO 54
      WRITE (12) 1,1,IX2,IY2
      GO TO 58
C
   54 WRITE (12) -1,1,IX1,IY1
      IX2 = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
      IY2 = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
      IF(IX2.LT.0 .OR. IX2.GT.9999 .OR. IY2.LT.0 .OR. IY2.GT.9999)
                                                                  GO TO 58
      GO TO 46
C
   58 CONTINUE
C
C         ON THE PLOT, CROSS OUT EITHER THE WORD FORECAST OR SOUNDING
C
   61 IF(IOPT(1) .NE. 0)GO TO 62
      WRITE (12) -1,1,707,604
      WRITE (12) 1,1,1174,604
      GO TO 64
C
   62 WRITE (12) -1,1,1269,604
      WRITE (12) 1,1,1760,604
C
C         PRINT OUT THE PREDICTION TIME ON THE PLOT
C
   64 WRITE (12) -1,1,1869,319
      WRITE (12,204) 100,0,0,150,ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
C
C         IF THE LAUNCH TIME WAS ENTERED, PRINT IT OUT ON THE PLOT
C
      IF(.NOT. LTIME)GO TO 67
      WRITE (12) -1,1,1869,1[illegible]
      WRITE (12,204) 100,0,0,150,LTIM,LAUNTD(4),LDAY,LMON,LYEAR
C
C         ON THE PLOT, PRINT OUT THE CHARACTERS + AND @ FOR THE LEGEND
```

```
C
   67 WRITE (12) -1,1,1041,1342
      WRITE (12,205) 150,0,0,150,1H+
      WRITE (12) -1,1,1041,1104
      WRITE (12,205) 150,0,0,150,1H@
C
C          FOR THE LEGEND ON THE PLOT, PRINT OUT THE CONCENTRATION VALUES
C          FOR WHICH CONTOURS WERE DRAWN
C
      WRITE (12) -1,1,1066,9587
      DO 75 I=1,10
      IF(CONC(I) .LT. 0.0)GO TO 77
      IF(I .NE. 1)GO TO 72
      WRITE (12,206) 125,0,0,150,CONC(I)
      GO TO 75
   72 WRITE (12,207) 125,0,0,150,CONC(I)
   75 CONTINUE
C
C          RETURN TO THE APPROPRIATE PLACE IN MOD3A
C
   77 GO TO IRETRN
   78 CALL MOD3A
C
C          END OF ISOPO
C
      END
      END$
```

Program REED

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONCLUDED)

Section Page

APPENDIX

LIST OF ILLUSTRATIONS

Figure | Page

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

```
FTN4,L
      PROGRAM REED
C
C************************************************************
C
C         NASA/MSFC MULTILAYER DIFFUSION MODEL -- 21 JUN 1977
C
C         MAIN PROGRAM -- REED
C
C************************************************************
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION ILINE(32),IDATAF(10),IERS(32),JJTIM(5)
C
C         INPUT FORMAT STATEMENTS
C
  100 FORMAT (I2,1X,2A2,I2)
  101 FORMAT (A1)
  102 FORMAT (10A2)
  103 FORMAT (I4,5XI2,1XA2,A1,1XI4)
  104 FORMAT (I4,3XI2,1XA2,A1,1XI4)
  105 FORMAT (I6,1XI3,1XF4.1,F6.1,F6.1,F7.2,11XF7.2)
  106 FORMAT (I4," C"A2,1XI2,1X,A2,A1,1X,I4)
  107 FORMAT (I2,1X,A2,A1,1X,I4)
  108 FORMAT (F7.2)
  109 FORMAT (I2)
  110 FORMAT (F4.1)
C
C         OUTPUT FORMAT STATEMENTS
C
  200 FORMAT ("1"80(1H*)/1X,80(1H*)/1X,10(1H*),60X,10(1H*)/
     .        1X,10(1H*)," NASA/MSFC MULTILAYER DIFFUSION MODEL - "
     .        "REED"4X"21 JUN 1977 ",10(1H*)/1X,10(1H*),60X,10(1H*)/
```

```
              1X,80(1H*)/1X,80(1H*)/
              "ORUN "I2" USING DATA FILE "3A2/
              "0"3A2,A1" LAUNCH VEHICLE")
  201 FORMAT ("OLAUNCH TIME: "I4,1XR1,A2,4X"DATE: "I2,1XA2,A1,1XI4)
  202 FORMAT ("OPREDICTION TIME: "I4" C"A2,4X"DATE: "I2,1XA2,A1,1XI4/
              "ODATA FILE HEADER INFORMATION:")
  203 FORMAT ("O<<<<OPEN ERROR "I4", PROCESSING CONTINUES WITH "
              "NEXT RUN>>>>")
  204 FORMAT ("O<<<<READF ERROR "I4", PROCESSING CONTINUES WITH "
              "NEXT RUN>>>>")
  205 FORMAT (6X,40A2)
  206 FORMAT ("O"5X"TIME: "I4,1XA1,A2,4X"DATE: "I2,1XA2,A1,1XI4)
  207 FORMAT ("1"80(1HS)/1X,20(1HS),40X,20(1HS)/
              1X,20(1HS),16X"SOUNDING"16X,20(1HS)/
              1X,20(1HS),40X,20(1HS)/1X,80(1HS)//)
  208 FORMAT ("1"80(1HF)/1X,20(1HF),40X,20(1HF)/
              1X,20(1HF),16X"FORECAST"16X,20(1HF)/
              1X,20(1HF),40X,20(1HF)/1X,80(1HF)//)
  209 FORMAT ("OSURFACE DENSITY (GM/M**3): "F8.2)
  210 FORMAT ("OLAYER   ALTITUDE      DIRECTION  SPEED   TEMP   "
              "POT-TEMP  D P TEMP  PRESSURE"/
              "  NO. (FEET) (METERS) (DEGREES) (M/SEC)     "
              "(DEGREES CENTIGRADE)    (MILLIBARS)")
  211 FORMAT (2XI2,I7,2XI5,7XI3,5XF4.1,4XF4.1,4XF5.2,6XF4.1,6XF5.2)
C
C        TYPE AND DIMENSION STATEMENTS
C
      INTEGER BLANKS,FILE(3),FDIGIT(50),RCHAR,TCHAR,SCHAR,
              VNAMES(4,3),RUNNUM,RA,FO,SDT,TE,ZEROO,RCLDR(3)
      DIMENSION IPAR(5),VPARS(18,5),IDCB(272),IBUF(40),IALT(31),
                DPTEMP(31),NAME(3),NAMEF(3)
C
C        DATA STATEMENTS
C
      DATA NAME/036522B,2HEE,1HD/
      DATA IERS/32*2H  /
      DATA NAMEF/2H?R,2HEE,1HD/
C
C***********************************************************************
C***  VPARS(1 THRU 18) = SHUTTLE    (19 - 36) = TITAN    (37 - 54)=DELTA
C***        (55 - 72) = DELTA 3914    (73 - 90) = MINUTEMAN
C***********************************************************************
C
      DATA VPARS/1.521923E7,6.882968E6,3.441484E6,1.894794173E9,
                 8.569295 16E8,1.713859032E9,.6522129891,.4680846,
                 .375,1479.7,1062.35,1000.0,.1970,.2234,.0316,.2791,
                 .0002,.64,
                 5.437528E6,2.718764E6,1.359382E6,3.2625168E8,
                 1.6312584E8,3.2625168E8,.429580469,.5184223,
                 5.0,2021.1,1010.55,1000.0,.1932,.2665,.0222,
     .          .2819,.0002,.64,
     .          8.360685E5,9.09811E4,2.729434E5,2.887598E7,
     .          3.14229E6,1.885373E7,.922156,.432703,.54,1766.0,
     .          1000.0,690.0,.1866,.2055,.0156,.3391,.0002,
     .          .50,
     .          1.057557E6,1.482923E5,3.70731E5,6.70269E7,
     .          9.398616E6,4.699308E7,1.245756,.4180947,
     .          0.0,1449.9,1000.0,411.18,.1866,.2055,.0156,.3391,
     .          .0002,.50,
     .          4.684476E5,4.684476E5,1.171119E5,2.8106856E7,
     .          2.8106856E7,2.8106856E7,.469982,.463333,0.0,
     .          2055.9,2055.9,1000.0,.1866,.2055,.0156,.3391,
     .          .0002,.64/
C
      DATA BLANKS/2H  /, RCHAR/1HR/, TCHAR/1HT/, SCHAR/1HS/,
     .     RA/2HRA/, FO/2HFO/, SDT/2HDT/, TE/2HTE/, ZERO0/2H00/,
     .     NINE9/2H99/, RCLDR/2HRC,2HLD,1HR/
      DATA FDIGIT/2H01,2H02,2H03,2H04,2H05,2H06,2H07,2H08,2H09,2H10,
     .            2H11,2H12,2H13,2H14,2H15,2H16,2H17,2H18,2H19,2H20,
     .            2H21,2H22,2H23,2H24,2H25,2H26,2H27,2H28,2H29,2H30,
     .            2H31,2H32,2H33,2H34,2H35,2H36,2H37,2H38,2H39,2H40,
     .            2H41,2H42,2H43,2H44,2H45,2H46,2H47,2H48,2H49,2H50/
      DATA VNAMES/2HSH,2HUT,2HTL,1HE,
     .            2HTI,2HTA,1HN,1H ,
     .            2HDE,2HLT,1HA,1H /
C
C        CALL GRAF TO INITIALIZE SCOPE (ONLY APPLICABLE IF USING
C        PLASMASCOPE)
C
      CALL GRAF(1)
C
C        FIND THE LOGICA' UNIT NUMBER OF THE DEVICE TO BE USED FOR
C        INPUT AND SET THE VARIABLE LU EQUAL TO IT
C
      CALL RMPAR(IPAR)
      LU = IPAR(1)
C
C        BEGIN PROCESSING OF NEW DATA BY CLEARING SCOPE
C
    1 CALL CLEAR
C
C        INITIALIZE SOME COMMON VARIABLES
C
      LTIME = .FALSE.
      YES = 1HY
      PI = 3.141593
      PIOVR2 = 0.5 * PI
      PI43 = 1.3333333 * PI
      TWOPI = 2.0 * PI
      SQR2PI = SQRT(TWOPI)
      DEGRAD = PI/180.0
      RADDEG = 180.0/PI
      DO 2 I=1,3
    2 IOPT(I) = 0
      JBOT = 0
      ZB = 0.0
      ZZ = 0.0
      REFLEC = 1.0
C
C         WRITE THE HEADER OF THE CONSOLE
C
      CALL CLEAR
      YPOS = 490.
      CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      CALL GETTD(LTIM,LDAY,LMON,LYEAR)
      CALL CODE
      WRITE (IDATAF,107) LDAY,LMON,LYEAR
      CALL CHAR(368.,YPOS,0,IDATAF,11,2,0)
      YPOS = YPOS - 32.
C
C         READ IN THE NUMBER OF RUNS TO BE MADE AND THE FIRST FOUR
C         CHARACTERS OF THE DATA FILE NAMES FOR THOSE RUNS
C
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,43,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(IX .GE. 28)IOPT(2) = 1
      IF(IOPT(2) .EQ. 0)GO TO 4
      CALL DREAD(NAMEF,4,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,22,3,0)
      NIN  = 9
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,175.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,22,0,0)
      YPOS = YPOS - 32.
      CALL CODE
      READ (IDATAF,100) NUMRUN,FILE(1),FILE(2),IFOFF
      NUMRUN = MIN0(MAX0(NUMRUN,1),50)
      IF(IFOFF .GT. 0)IFOFF = IFOFF - 1
      IF(FILE(1) .NE. BLANKS)GO TO 5
    4 FILE(1) = 2HDA
      FILE(2) = 2HTA
      IFOFF = 0
      NUMRUN = 1
    5 IF(NUMRUN+IFOFF .GT. 50)NUMRUN = 50 - IFOFF
      IFOFF1 = IFOFF + 1
      IFLAG(3) = 0
      IF(FILE(1).EQ.2HVA .AND. FILE(2).EQ.2HND)IFLAG(3) = 1
      IF(FILE(1).EQ.2HTA .AND. FILE(2).EQ.2HPE)IFLAG(3) = 2
      IF(FILE(1).EQ.2HDA .AND. FILE(2).EQ.2HTA)IFLAG(3) = 3
      IFLAG(4) = 1HE
      IF(IFLAG(3) .EQ. 1)IFLAG(4) = 1HP
C
C         FIND OUT IF THESE RUNS ARE TO BE RESEARCH RUNS (INTERACTION
C         AND PLOTTING ALLOWED) OR PRODUCTION RUNS
C
      CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,11,3,0)
      CALL CHAR(128.,YPOS,0,ILINE(9),12,3,0)
      CALL CHAR(240.,YPOS,0,ILINE(16),32,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LT. 12)CALL CHAR(224.,YPOS,0,IERS,34,0,0)
      IF(IX .LT. 12)IFLAG(1) = 1
      IF(IX.GE.12 .AND. IX.LT.19)CALL CHAR(120.,YPOS,0,IERS,16,0,0)
      IF(IX.GE.12 .AND. IX.LT.19)IFLAG(1) = 2
      IF(IX.GE.12 .AND. IX.LT.19)CALL CHAR(368.,YPOS,0,IERS,16,0,0)
      IF(IX .GE. 19)CALL CHAR(120.,YPOS,0,IERS,30,0,0)
      IF(IX .GE. 19)IFLAG(1) = 3
      YPOS = YPOS - 32.
      IF(IX .LT. 19)IOPT(2) = 1
      IF(IOPT(2) .EQ. 0)GO TO 7
      GO TO 12
    7 CONTINUE
C
C         FOR PRODUCTION RUNS, READ IN THE TOP OF THE SURFACE LAYER
C         AND THE SIGMA OF THE WIND AZIMUTH ANGLE TO BE USED FOR ALL RUNS
C
      CALL DREAD(NAMEF,11,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,33,3,0)
      NIN = 6
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,263.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,33,0,0)
      CALL CODE
      READ (IDATAF,108) TOPSUR
      YPOS = YPOS - 32.
      CALL DREAD(NAMEF,12,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,37,3,0)
      NIN = 2
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,295.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,37,0,0)
      CALL CODE
      READ (IDATAF,109) ISIGA
      IF(NIN.EQ.1) ISIGA = ISIGA/10
      SIGA = FLOAT(ISIGA)
      YPOS = YPOS - 32.
C
C        READ IN AND WRITE OUT THE LAUNCH TIME AND DATE -- IF NOT
C        ENTERED, DO NOT WRITE ANYTHING OUT
C
   12 CALL DREAD(NAMEF,8,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,28,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL GETTD(LTIM,LDAY,LMON,LYEAR)
      CALL CODE
      WRITE (IDATAF,106) LTIM,SDT,LDAY,LMON,LYEAR
      CALL CHAR(224.,YPOS,0,IDATAF,20,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,28,0,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),15,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(IX .LE. 25)GO TO 17
      CALL DREAD(NAMEF,9,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,26,3,0)
      NIN = 20
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,207.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,26,0,0)
      CALL CODE
      READ (IDATAF,102) (LAUNTD(I),I=1,10)
      YPOS = YPOS - 32.
      IF(LAUNTD(1) .EQ. BLANKS)GO TO 17
      LTIME = .TRUE.
      CALL CODE
      READ (LAUNTD,103) LTIM,LDAY,LMON,LYEAR
      GO TO 21
   17 LAUNTD(4) = SDT
C
C        READ IN THE LAUNCH VEHICLE, LET IT DEFAULT IF NOT ENTERED,
C        WRITE IT BACK OUT, AND FILL THE VPAR ARRAY WITH THE
C        APPROPRIATE VEHICLE PARAMETERS
C
   21 CALL DREAD(NAMEF,10,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,24,3,0)
      CALL CHAR(192.,YPOS,0,ILINE(13),24,0,0)
      CALL CHAR(416.,YPOS,0,ILINE(27),11,3,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 15)CALL CHAR(312.,YPOS,0,IERS,24,0,0)
      IF(IX.GT.15 .AND. IX.LT.24)CALL CHAR(192.,YPOS,0,IERS,12,0,0)
      IF(IX.GT.15 .AND. IX.LT.24)CALL CHAR(416.,YPOS,0,IERS,11,0,0)
      IF(IX .GE. 24)CALL CHAR(192.,YPOS,0,IERS,24,0,0)
      YPOS = YPOS - 32.
      IF(IX .LE. 15)GO TO 25
      IF(IX .LE. 23)GO TO 24
C*******************************************************************************
C******     IF IOPT(3) = 0  THEN IT IS A SHUTTLE LAUNCH.                 *******
C*******************************************************************************
      IOPT(3) = 0
      GO TO 26
C*******************************************************************************
C******    IF IOPT(3) = 1   THEN IT IS A TITAN LAUNCH.                   *******
C*******************************************************************************
   24 IOPT(3) = 1
      GO TO 26
C*******************************************************************************
C******     IF IOPT(3) = 2   THEN IT IS A DELTA LAUNCH.                  *******
C*******************************************************************************
   25 IOPT(3) = 2
   26 I = IOPT(3) + 1
C
C        FILL THE VPAR ARRAY
C
      DO 28 J=1,18
   28 VPAR(J) = VPARS(J,I)
C
C        CHANGE IN BOTTOM LAYER WITH TOTAL REFLECTION?
C
      CALL DREAD(NAMEF,15,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 32.
      CALL DREAD(NAMEF,16,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(32.,YPOS,0,ILINE(3),11,3,0)
      CALL CHAR(160.,YPOS,0,ILINE(11),43,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      YP = YPOS
      YPOS = YPOS - 32.0
C
C         CHECK FOR SURFACE -- STABILIZATION -- SOMETHING ELSE
C
      IF(IX .GE. 8)GO TO 29
      IFLAG(2) = 0
      CALL CHAR(0.0,YP,0,ILINE,16,0,0)
      GO TO 38
   29 IF(IX .GE. 20)GO TO 30
      IFLAG(2) = 1
      CALL CHAR(160.0,YP,0,ILINE(11),16,0,0)
      JBOT = 2
      ZB = ALT(JBOT)
      GO TO 38
C
C         DEFAULT HEIGHT CALCULATION Zt?
C
   30 IFLAG(2) = 2
      CALL CHAR(320.0,YP,0,ILINE(20),18,0,0)
      CALL DREAD(NAMEF,17,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,42,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,IERS,42,0,0)
      CALL CHAR(0.,YPOS,0,ILINE,42,0,0)
      IF(IX .LE. 25)GO TO 37
      CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
C
C         ENTER HEIGHT Zz
C
      CALL DREAD(NAMEF,18,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(47.,YPOS,0,ILINE(4),10,3,0)
      NIN = 6
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,128.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) ZZ
      CALL CHAR(0.,YPOS,0,IERS,16,0,0)
      CALL CHAR(47.,YPOS,0,ILINE(4),10,0,0)
      YPOS = YPOS - 32.
C
C         ENTER SURFACE REFLECTION?
```

```
C
      CALL DREAD(NAMEF,19,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,45,3,0)
      CALL CHAR(383.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(472.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)REFLEC = 1.0
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .LE. 25)GO TO 34
      CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
C
C         WRITE OUT Rf VALUES FOR SELECTION
C
      CALL DREAD(NAMEF,20,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,3,0)
      NIN = 4
      CALL BLANK(IDATAF,10)
      CALL IN(2,JTYPE,440.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      IF(JTYPE .NE. 0)GO TO 31
      CALL CODE
      READ (IDATAF,*) REFLEC
      GO TO 32
   31 IX = IX/2
      IF(IX .EQ. 1)REFLEC = 0.8
      IF(IX .EQ. 3)REFLEC = 0.7
      IF(IX .EQ. 5)REFLEC = 0.5
      IF(IX .EQ. 7)REFLEC = 0.3
      IF(IX .EQ. 9)REFLEC = 0.1
      IF(IX .EQ. 11)REFLEC = 0.0
      CALL CODE
      WRITE (IDATAF,110) REFLEC
   32 CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(48.,YPOS,0,ILINE,6,0,0)
      CALL CHAR(88.,YPOS,0,IDATAF,4,0,0)
      IF(JTYPE .NE. 0)GO TO 34
      CALL CODE
      READ (IDATAF,*) REFLEC
   34 YPOS = YPOS - 32.
C
C         DEFAULT HEIGHT OF BASE LAYER?
C
      CALL DREAD(NAMEF,21,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,46,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .LE. 25)GO TO 36
      CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      CALL DREAD(NAMEF,22,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(47.,YPOS,0,ILINE(4),10,3,0)
C
C        ENTER HEIGHT OF BASE LAYER
C
      NIN = 6
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,144.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) ZB
      CALL CHAR(47.,YPOS,0,ILINE(4),10,0,0)
      YPOS = YPOS - 32.
      GO TO 38
   36 ZB = 0.0
      GO TO 38
   37 CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.,YPOS,0,ILINE,58,0,0)
      YPOS = YPOS - 32.
   38 CONTINUE
C
C        DO LOOP ON THE RUN NUMBER
C
      DO 79 RUNNUM=1,NUMRUN
C
C        SET UP THE FILE NAME FOR THIS RUN, GET THE CURRENT TIME,
C        AND WRITE OUT THE HEADER
C
      FILE(3) = FDIGIT(RUNNUM+IFOFF)
      CALL GETTD(ITIME,IDAY,MONTH,IYEAR)
      I = IOPT(3) + 1
      WRITE (6,200) RUNNUM,(FILE(J),J=1,3),(VNAMES(J,I),J=1,4)
      IF(LTIME)WRITE (6,201) LTIM,LAUNTD(3),LAUNTD(4),LDAY,LMON(1),
     *                        LMON(2),LYEAR
      WRITE (6,202) ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
C
C        IF THE DATA IS ON A DISK FILE, READ FROM DISK -- IF IT
C        IS ON TAPE, READ IT AS KSC 1965 DATA IN SUBROUTINE KSC65
C
      IF(IFLAG(3) .NE. 2)GO TO 39
      CALL KSC65(IBUF,IALT,DPTEMP,IFOFF1,IEOF)
      IFOFF1 = 1
      IF(IEOF .EQ. 1)GO TO 81
      IF(IEOF .EQ. 2)GO TO 79
      GO TO 48
C
C        OPEN THE DATA FILE FOR THIS RUN
C
   39 CALL OPEN(IDCB,IERR,FILE,0,0,0,272)
      IF(IERR .GE. 0)GO TO 40
      WRITE (6,203) IERR
      GO TO 79
C
C        READ THE HEADINGS FROM THE DATA FILE, SETTING UP THE
C        APPROPRIATE PARAMETERS
C
   40 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .GE. 0)GO TO 42
   41 WRITE (6,204) IERR
      GO TO 79
   42 IF(IBUF(1) .NE. TE)GO TO 40
   43 WRITE (6,205) (IBUF(I),I=1,LEN)
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      IF(IBUF(1).NE.RA .AND. IBUF(1).NE.FO)GO TO 43
      IOPT(1) = 0
      IF(IBUF(1) .EQ. FO)IOPT(1) = 1
      WRITE (6,205) (IBUF(I),I=1,LEN)
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      WRITE (6,205) (IBUF(I),I=1,LEN)
C
C        READ THE SOUNDING/FORECAST TIME
C
      CALL READF(IDCB,IERR,IBUF,9)
      IF(IERR .LT. 0)GO TO 41
      CALL CODE
      READ (IBUF,104) ISTIM,ISDAY,ISMON(1),ISMON(2),ISYEAR
C
C        CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME
C
      ISTIM = ISTIM - 500
      IF(IFLAG(3) .EQ. 1)ISTIM = ISTIM - 300
      IF(LAUNTD(4) .NE. 2HST)ISTIM = ISTIM + 100
      IF(ISTIM .GT. 0)GO TO 44
      ISTIM = 2400 + ISTIM
      ISDAY = ISDAY - 1
C
C        WRITE OUT THE NEXT LINE OF THE HEADER
C
   44 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      WRITE (6,205) (IBUF(I),I=1,LEN)
C
```

```
C          WRITE OUT THE SOUNDING/FORECAST TIME
C
      WRITE (6,206) ISTIM,IFLAG(4),LAUNTD(4),ISDAY,ISMON(1),ISMON(2),
     *              ISYEAR
C
C          FIND THE FIRST DATA POINT WITH AN ALTITUDE OF 10 FEET
C          OR ABOVE
C
   45 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      CALL B2Z(IBUF(1),J)
      IF(J.LT.ZERO0 .OR. J.GT.NINE9)GO TO 45
      CALL CODE
      READ (IBUF,105) IALT(1),IDIR(1),SPEED(1),TEMP(1),DPTEMP(1),
     *                PRESS(1),SURDEN
      IF(IALT(1) .LT. 10)GO TO 45
C
C          TRY TO FIND A TOTAL OF 30 DATA POINTS WITH ALTITUDES
C          BETWEEN 20 FT AND 10,000 FT INCLUSIVE
C
      NUM = 1
      DO 47 I=2,30
   46 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0 .AND. IERR.NE.-12)GO TO 41
      IF(LEN .EQ. -1)GO TO 48
      CALL B2Z(IBUF(1),J)
      IF(J.LT.ZERO0 .OR. J.GT.NINE9)GO TO 46
      CALL CODE
      READ (IBUF,105) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),
     *                 PRESS(I)
      IF(IALT(I).LT.20 .OR. IALT(I).GT.10000)GO TO 46
   47 NUM = I
C
C          ZERO OUT THE REMAINING ELEMENTS OF THE ARRAYS
C
   48 NUM1 = NUM + 1
      IF(NUM1 .GT. 30)GO TO 51
      DO 49 I=NUM1,30
      ALT(I) = 0.0
      IDIR(I) = 0
      SPEED(I) = 0.0
      TEMP(I) = 0.0
      DPTEMP(I) = 0.0
   49 PRESS(I) = 0.0
C
C          CONVERT TO METRIC UNITS
C
   51 DO 52 I=1,NUM
      ALT(I) = 0.3048 * FLOAT(IALT(I))
   52 SPEED(I) = 0.515 * SPEED(I)
```

```
C
C          SORT ALL THE DATA POINTS SO THEY APPEAR IN ASCENDING
C          ORDER OF ALTITUDE
C
      NUM1 = NUM - 1
      DO 58 I=1,NUM1
      JJ = NUM - I
      DO 57 J=1,JJ
      J1 = J + 1
      IF(ALT(J) .LE. ALT(J1))GO TO 57
      ARG = ALT(J)
      ALT(J) = ALT(J1)
      ALT(J1) = ARG
      IARG = IDIR(J)
      IDIR(J) = IDIR(J1)
      IDIR(J1) = IARG
      ARG = SPEED(J)
      SPEED(J) = SPEED(J1)
      SPEED(J1) = ARG
      ARG = TEMP(J)
      TEMP(J) = TEMP(J1)
      TEMP(J1) = ARG
      ARG = DPTEMP(J)
      DPTEMP(J) = DPTEMP(J1)
      DPTEMP(J1) = ARG
      ARG = PRESS(J)
      PRESS(J) = PRESS(J1)
      PRESS(J1) = ARG
   57 CONTINUE
   58 CONTINUE
C
C          CALCULATE THE POTENTIAL TEMPERATURE
C
      DO 62 I=1,NUM
   62 PTEMP(I) = (TEMP(I) + 273.15) * ((1000.0/PRESS(I))**0.288)
C
C          WRITE THE HEADER FOR SOUNDING OR FORECAST
C
      IF(IOPT(1) .EQ. 1)GO TO 64
      WRITE (6,207)
      GO TO 65
   64 WRITE (6,208)
C
C          WRITE THE SURFACE DENSITY AND ALL THE DATA POINTS
C
   65 WRITE (6,209) SURDEN
      WRITE (6,210)
      DO 68 I=1,NUM
      IALTF = 3.281 * ALT(I) + 0.5
      IALTM = ALT(I) + 0.5
      APTEMP = PTEMP(I) - 273.15
   68 WRITE (6,211) I,IALTF,IALTM,IDIR(I),SPEED(I),TEMP(I),
     +                 APTEMP,DPTEMP(I),PRESS(I)
C
C         CLOSE THE DATA FILE
C
      CALL CLOSE(IDCB)
C
C         TRANSFER TO THE PROGRAM RCLDR -- THE CLOUD RISE PROGRAM
C
      CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RCLDR)
      CALL RWDIS(NAME,0)
      CALL GRAF(1)
C
C         PROCESS THE NEXT RUN
C
   79 CONTINUE
C
C         TERMINATE OF PROCESS MORE DATA?
C
   81 CALL DREAD(NAMEF,14,ILINE)
      CALL LEPS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE(1),24,3,0)
      CALL CHAR(224.,YPOS,0,ILINE(15),14,3,0)
      CALL CHAR(352.,YPOS,0,ILINE(23),12,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      IF(IX .LT. 20)GO TO 82
C
C         PROCESS MORE DATA
C
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE(1),28,0,0)
      CALL CHAR(352.,YPOS,0,ILINE(23),12,0,0)
      YPOS = 458.
      GO TO 1
C
C         TERMINATE EXECUTION
C
   82 CALL DREAD(NAMEF,13,ILINE)
      CALL LERS(YPOS)
      YPOS = YPOS - 32.
      CALL CHAR(0.,YPOS,0,ILINE,64,3,0)
C
C         DELAY BEFORE CLEARING SCOPE
C
      CALL EXEC(11,JJTIM)
      JJ=JJTIM(2)
      IF(JJ.GT.55) JJ =5
   85 CALL EXEC(11,JJTIM)
      IF(JJ+5 .GT. JJTIM(2))GO TO 85
C
C          REINITIALIZE SCOPE NORMAL OPERATION AND STOP
C
      CALL NGRAF
      STOP
C
C          END OF REED
C
      END
      SUBROUTINE KSC65(IBUF,IALT,DPTEMP,IWANT,IEOF)
C
C*********************************************************
C
C          THIS SUBROUTINE READS IN DATA FOR THE REED DIFFUSION
C          MODEL FROM MAG TAPE IN KSC 1965 FORMAT
C
C*********************************************************
C
C          COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,P1,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C          INPUT FORMAT STATEMENTS
C
 1000 FORMAT (40A2)
 1001 FORMAT (I4,3XI2,1XA2,A1,1XI4)
 1002 FORMAT (1XI6,3XI3,5XF3.0,2XF5.1,3XF5.1,3XF6.1,15XF6.1)
C
C          OUTPUT FORMAT STATEMENT
C
 2000 FORMAT ("0"5X"TIME: "I4,1XA1,A2,4X"DATE: "I2,1XA2,A1,1XI4)
C
C          DIMENSION STATEMENT
```

```
C
      DIMENSION IBUF(1),IALT(1),DPTEMP(1)
C
C        INITIALIZE THE COUNTER FOR THE NUMBER OF SETS OF DATA TO 0
C
      IGOT = 0
C
C        READ DATA FROM TAPE
C
    4 READ (8,1000) (IBUF(I),I=1,40)
C
C        IF AN EOF ON TAPE, SET THE EOF FLAG AND RETURN
C
      CALL EXEC(13,8,IEQT5)
      IEOF = IAND(ISHIF(IEQT5,-7),1)
      IF(IEOF .EQ. 1)RETURN
C
C        KEEP READING UNTIL THE STANDARD LEVEL DATA IS FOUND
C
      IF(IBUF(2) .NE. 2HST)GO TO 4
    7 READ (8,1000) (IBUF(I),I=1,40)
      IF(IBUF(1).NE.2HCA .OR. IBUF(2).EQ.2HST)GO TO 7
C
C        READ THE SOUNDING/FORECAST TIME
C
      READ (8,1001) ISTIM,ISDAY,ISMON(1),ISMON(2),ISYEAR
C
C        CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME
C
      ISTIM = ISTIM - 500
      IF(IFLAG(3) .EQ. 1)ISTIM = ISTIM - 300
      IF(LAUNTD(4) .NE. 2HST)ISTIM = ISTIM + 100
      IF(ISTIM .GT. 0)GO TO 11
      ISTIM = 2400 + ISTIM
      ISDAY = ISDAY - 1
C
C        FIND THE KEY WORD ALTITUDE
C
   11 READ (8,1000) (IBUF(I),I=1,40)
      IF(IBUF(2) .EQ. 2HST)GO TO 7
      IF(IBUF(1) .NE. 2HAL)GO TO 11
C
C        LIMIT DATA TO 30 POINTS -- READ THE STANDARD LEVEL DATA
C
      DO 19 I=1,30
   15 READ (8,1002) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),PRESS(I),
                   SURDN
      IF(SPEED(I).EQ.999.0 .OR. IDIR(I).EQ.999)GO TO 15
      IF(IDIR(I) .EQ. 360)IDIR(I) = 0
      IF(I .EQ. 1)SURDEN = SURDN
      IF(IALT(I) .GT. 10000)GO TO 22
   19 CONTINUE
   22 NUM = I
      IF(NUM .GT. 30)GO TO 34
C
C        FIND THE KEY WORD MANDATORY
C
   25 READ (8,1000) (IBUF(I),I=1,40)
      IF(IBUF(2) .EQ. 2HST)GO TO 7
      IF(IBUF(10).NE.2HOR .AND. IBUF(15).NE.2HOR)GO TO 25
      READ (8,1000) I
C
C        LIMIT DATA TO 30 POINTS -- READ THE MANDATORY LEVEL DATA
C
      DO 29 I=NUM,30
   27 READ (8,1002) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),PRESS(I)
      IF(SPEED(I).EQ.999.0 .OR. IDIR(I).EQ.999)GO TO 27
      IF(IDIR(I) .EQ. 360)IDIR(I) = 0
      IF(IALT(I) .GT. 10000)GO TO 34
   29 CONTINUE
C
C        NUM IS THE NUMBER OF DATA POINTS
C
   34 NUM = I - 1
C
C        INCREMENT THE COUNTER -- IF THIS IS THE SET OF DATA DESIRED,
C        WRITE OUT THE SOUNDING/FORECAST TIME -- OTHERWISE GET THE NEXT
C        SET
C
      IGOT = IGOT + 1
      IF(IGOT .LT. IWANT)GO TO 4
C
C        WRITE OUT THE SOUNDING/FORECAST TIME
C
      WRITE (6,2000) ISTIM,IFLAG(4),LAUNTD(4),ISDAY,ISMON(1),ISMON(2),
     *               ISYEAR
C
C        THERE MUST BE 5 OR MORE DATA POINTS FOR THIS TO BE A VALID SET
C        OF DATA -- IF THERE IS NOT, RETURN WITH IEOF=2
C
      IF(NUM .GE. 5)RETURN
      IEOF = 2
      RETURN
C
C        END OF KSC65
C
      END
      SUBROUTINE RWDIS(NAME,JJ)
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     *       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
            ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
            LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
            SIGHCL,RADDEG,R,TOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
            SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
            TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(3,),DIST,YES,Y1,NUMRUN,
            YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
               (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
               (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
               (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
               (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
               (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER ODCB(144),OBUF(669)
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),ALT(1))
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,669)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,669)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE GETTD(ITIME,IDAY,MONTH,IYEAR)
C
C*********************************************************************
C
C        THIS SUBROUTINE RETURNS THE CURRENT TIME, DAY, MONTH, AND YEAR
C
C*********************************************************************
C
C
C        TYPE AND DIMENSION STATEMENTS
C
      INTEGER DAYMON(12)
      DIMENSION MONTH(2),MONTHS(2,12),IT(5)
C
C        DATA STATEMENTS
C
      DATA MONTHS/2HJA,1HN,2HFE,1HB,2HMA,1HR,2HAP,1HR,
               2HMA,1HY,2HJU,1HN,2HJU,1HL,2HAU,1HG,
               2HSE,1HP,2HOC,1HT,2HNO,1HV,2HDE,1HC/
      DATA DAYMON/31,28,31,30,31,30,31,31,30,31,30,31/
C
C        CALL EXEC TO RETURN CURRENT TIME, JULIAN DAY, AND YEAR
C
      CALL EXEC(11,IT,IYE[illegible]
C
C        USE JUST HOURS AND MINUTES FOR THE TIME
C
```

```
      ITIME = 100 * IT(4) + IT(3)
C
C         MAKE APPROPRIATE ADJUSTMENTS IF THIS IS A LEAP YEAR
C
      DAYMON(2) = 28
      I = IYEAR/4
      IF(4*I .EQ. IYEAR)DAYMON(2) = 29
C
C         CONVERT THE JULIAN DAY INTO A MONTH AND A DAY
C
      IDAY = IT(5)
      DO 7 I=1,12
      IDAY = IDAY - DAYMON(I)
      IF(IDAY .LE. 0)GO TO 12
    7 CONTINUE
   12 IDAY = IDAY + DAYMON(I)
      MONTH(1) = MONTHS(1,I)
      MONTH(2) = MONTHS(2,I)
C
C         RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C         END OF GETTD
C
      END
      SUBROUTINE B2Z(IA,IB)
      IB = IAND(IA,177400B)
      IF(IB .EQ. 020000B)IB = 030000B
      IC = IAND(IA,000377B)
      IF(IC .EQ. 000040B)IC = 000060B
      IB = IOR(IB,IC)
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
      DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32),IPAR(5)
      CALL RMPAR(IPAR)
      LU = IPAR(1)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
10    CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
100   FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
10    IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS)
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS.LE.48) YPOS = 458.0
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.,YPOS-16.,0,IERS,64,0,0)
      RETURN
      END
      END$
```

```
FTN4,L
      PROGRAM RCLDR
C
C**********************************************************************
C
C         CLOUD RISE PROGRAM -- A PROGRAM IN THE REED SERIES OF
C         PROGRAMS
C
C**********************************************************************
C
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLORAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN, IGZ0,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
      INTEGER RMETP(3)
      DIMENSION IAS(31),NAME(3),NAMEF(3),ILINE(32),IDATAF(10),
                IERS(80),ISURTP(3)
C
C         OUTPUT FORMAT STATEMENTS
C
  100 FORMAT (F7.2)
  101 FORMAT (I2)
102   FORMAT(F3.1)
  200 FORMAT ("1"27X"EXHAUST CLOUD"/"0LEVEL"4X"ALTITUDE"17X
              "RISE TIME"5X"RANGE"6X"DIRECTION"/10X"(METERS)"17X
              "(SECONDS)"4X"(METERS)"4X"(DEGREES)")
  201 FORMAT (2XI3,5XF7.1,5X"ADIABATIC"5XF6.1,6XF7.1,7XF5.1)
  202 FORMAT (2XI3,5XF7.1,6X"STABLE"7XF6.1,6XF7.1,7XF5.1)
  203 FORMAT (//"0****CLOUD STABLIZATION****"/
              6X"HEIGHT(M): "F6.1/
              6X"STABILIZATION TIME AFTER LAUNCH(SEC): "F5.1/
              6X"RANGE FROM PAD(M): "F7.1/
              6X"DIRECTION FROM PAD(DEG): "F5.1)
  204 FORMAT (F6.1)
  205 FORMAT (//"0****TOP OF SURFACE LAYER METEOROLOGICAL PARAMETERS"
               "****"/
               6X"HEIGHT(M): "F6.1/
               6X"WIND DIRECTION(DEG): "I3/
               6X"WIND SPEED(M/SEC): "F4.1)
  206 FORMAT (//"0****DIFFUSION PARAMETERS****"/
               6X"MEAN SPEED(M/SEC): "F4.1/
               6X"MEAN TRANSPORT DIRECTION(DEG): "F5.1)
  207 FORMAT (F3.0)
  208 FORMAT (//"0SIGMA OF WIND AZIMUTH ANGLE, SIGA: "F4.1)
  209 FORMAT (//"0EFFECTIVE CLOUD HEIGHT(M): "F6.1)
C
C          TYPE AND DIMENSION STATEMENTS
C
      INTEGER RCONC(3)
C
C          DATA STATEMENT
C
      DATA NAME/036522B,2HEE,1HD/,NAMEF/2HPR,2HCL,2HDR/
      DATA RMETP/2HRM,2HET,1HP/
      DATA RCONC/2HRC,2HON,1HC/
      DO 1 I=1,80
    1 IERS(I) = 2H
C
C  ** CALL GRAF(1) TO INITIALIZE PLASMASCOPE GRAPHIC MODE
      CALL GRAF(1)
C  ** INITRIALIZE THE Y POSITION OF THE CALL CHARACTER STATEMENTS
C  ** ON THE PLASMASCOPE.
C     YPOS=490.
C  *** READ COMMON DATA FILE ***
      CALL RWDIS(NAME,0)
C
C          INITIALIZE SOME LOCAL VARIABLES
C
C          CRTIME( ) - CLOUD RISE TIME
C          IAS( ) - 0 = ADIABATIC
C                   1 = STABLE
C          ALTINC - ALTITUDE INCREMENT
C          ITERAT - ITERATION COUNTER
C
      RNGY = 0.0
      RNGX = 0.0
      CRTIME(1) = 0.0
      ALTINC = 0.0
      SAVER(1) = 0.0
      SAVEA(1) = 0.0
      ITERAT = 0
C
C          WRITE OUT THE EXHAUST CLOUD HEADER
C
```

```
      WRITE (6,200)
C
C        CALCULATE SOME QUANTITIES TO BE USED IN SUBSEQUENT DO LOOP
C
      ALPHAC = 5.12913086E-02*(TEMP(1) + 273.15)*SURDEN*GAMMAX**3
      ALPHAC = ALPHAC/(HEATN * QC1)
      GT = 9.8/(TEMP(1) + 273.15)
C
C        DO LOOP TO CALCULATE EXHAUST CLOUD PARAMETERS
C
      DO 9 I=2,NUM
C
      IM1 = I - 1
      IAS(I) = 1
C
C        CALCULATE SLOPE OF POTENTIAL TEMPERATURE, SPEED, AND
C        DIRECTION IN LAYER
C
      DALT = ALT(I) - ALT(IM1)
      GPTEMP = (PTEMP(I) - PTEMP(IM1))/DALT
      GSPEED = (SPEED(I) - SPEED(IM1))/DALT
      GDIR = FLOAT(IDIR(I) - IDIR(IM1))/DALT
C
C        CALCULATE METEOROLOGICAL AND ENERGY FACTOR
C
    2 Z = ALT(I) - ALT(1) - ALTINC
      ALPHA = ALPHAC * Z**4/(AA * Z**BB + CC)
C
C        CALCULATE POTENTIAL TEMPERATURE FACTOR
C
      STAB = GT * (PTEMP(I) - ALTINC * GPTEMP - PTEMP(1))/
     1            (ALT(I) - ALTINC - ALT(1) + 1.0E-7)
C
C        CALCULATION FOR ADIABATIC RISE
C
      IF(STAB .GT. 0.000001)GO TO 4
      CRTIME(I) = SQRT(ALPHA)
      IAS(I) = 0
      GO TO 6
C
C        CALCULATION FOR STABLE CLOUD RISE
C
C        C2 - ARGUMENT OF ARC COSINE (MUST BE LESS THAN -1)
C
    4 C2 = 1.0 - 0.5 * ALPHA * STAB
      IF(C2 .LT. -1.0)GO TO 5
      C3 = C2/SQRT(1.0 - C2 * C2)
      CRTIME(I+ITERAT) = (PIOVR2 - ATAN(C3))/SQRT(STAB)
      IF(ITERAT .EQ. 1)GO TO 11
      GO TO 6
```

```
C
C          ITERATE IN LAYER
C
    5 ALTINC = ALTINC + 5.0
      ITERAT = 1
      GO TO 2
C
C          CALCULATE RANGE AND DIRECTION
C
    6 DELRNG = - 0.5 * (SPEED(IM1) + SPEED(I)) *
               (CRTIME(IM1) - CRTIME(I))
      DELDIR = 0.00872665 * FLOAT(IDIR(I) + IDIR(IM1))
      RNGY = RNGY - DELRNG * SIN(DELDIR)
      RNGX = RNGX - DELRNG * COS(DELDIR)
      AZMUTH = RADDEG * ATAN2(RNGY,RNGX)
      IF(AZMUTH .LT. 0.0)AZMUTH = AZMUTH + 360.0
      DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
      SAVER(I) = DELRNG
      SAVEA(I) = AZMUTH
C
C          WRITE OUT THE VARIABLES WITH THE APPROPRIATE FORMAT STATEMENT
C          BASED OF WHETHER OR NOT CLOUD IS ADIABATIC OR STABLE
C
      IF(IAS(I) .NE. 0)GO TO 8
      WRITE (6,201) I,ALT(I),CRTIME(I),DELRNG,AZMUTH
      GO TO 9
    8 WRITE (6,202) I,ALT(I),CRTIME(I),DELRNG,AZMUTH
    9 CONTINUE
C
C          CALCULATE AND WRITE OUT STABILIZATION HEIGHT AND TIME
C
   11 DELRNG = 0.5 * (SPEED(IM1) - ALTINC * GSPEED + SPEED(I)) *
                (CRTIME(I + 1) - CRTIME(IM1))
      DALT = 0.00872665 * (FLOAT(IDIR(I) + IDIR(IM1)) - GDIR * ALTINC)
      RNGY = RNGY - DELRNG * SIN(DALT)
      RNGX = RNGX - DELRNG * COS(DALT)
      AZMUTH = RADDEG * ATAN2(RNGY,RNGX)
      IF(AZMUTH .LT. 0.0)AZMUTH = AZMUTH + 360.0
      DELRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
      ALT(31) = ALT(I) - ALTINC
      WRITE (6,203) ALT(31),CRTIME(I+1),DELRNG,AZMUTH
C
C          STORE THE INDEX OF THE ESTIMATED TOP OF THE SURFACE LAYER
C
      JTOP = I
C
C          LOAD THE CLOUD RISE TIME ARRAY
C
      CRTIME(31) = CRTIME(JTOP)
      DO 15 J=I,NUM
   15 CRTIME(I) = CRTIME(31)
C
C          IS THIS A RESEARCH OR A PRODUCTION RUN?
C
      IF(IOPT(2) .NE. 0)GO TO 21
C
C          PRODUCTION RUN -- IF TOPSUR IS UNDEFINED, USE JTOP AS ESTIMATED
C
   17 IF(TOPSUR .LE. 0.0)GO TO 26
C
C          CALCULATE JTOP BASED ON VALUE OF TOPSUR
C
      LEASTD = 9999999.9
      DO 19 I=1,NUM
      DIFF = ABS(ALT(I) - TOPSUR)
      IF(DIFF .GT. LEASTD)GO TO 19
      LEASTD = DIFF
      JTOP = I
   19 CONTINUE
      GO TO 26
C
C          WRITE OUT THE ESTIMATED TOP OF SURFACE LAYER -- READ IN
C          THE ONE TO BE USED -- CALCULATE JTOP
C
   21 CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      CALL CODE
      WRITE (ISURTP,204) ALT(JTOP)
      TOPSUR = ALT(JTOP)
      CALL CHAR(320.,YPOS,0,ISURTP,6,0,0)
      YPOS = YPOS - 32.
      IF(IFLAG(1) .EQ. 3)GO TO 26
      IF(IFLAG(1) .EQ. 1)GO TO 24
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,6,3,0)
      CALL CHAR(56.,YPOS,0,IERS,1,3,0)
      CALL CHAR(64.,YPOS,0,ILINE(5),9,3,0)
      CALL CHAR(160.,YPOS,0,ILINE(11),44,0,0)
      NIN=6
      CALL BLANK(IDATAF,10)
      CALL IN(2,JTYPE,463.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      IF(JTYPE .EQ. 1)GO TO 22
      CALL CHAR(0.,YPOS,0,ILINE,6,0,0)
      CALL CHAR(47.,YPOS,0,IERS,40,0,0)
      YPOS=YPOS-32.
      CALL CODE
      READ (IDATAF,100) TOPSUR
      ALT(JTOP) = TOPSUR
      GO TO 17
   22 IF(IX .GT. 9)GO TO 23
      CALL CHAR(0.,YPOS,0,ILINE,18,0,0)
      CALL CHAR(143.,YPOS,0,IERS,46,0,0)
      YPOS = YPOS - 32.
      GO TO 17
   23 CALL CHAR(0.,YPOS,0,ILINE,6,0,0)
      CALL CHAR(56.,YPOS,0,IERS,10,0,0)
      CALL CHAR(360.,YPOS,0,IERS,18,0,0)
      YPOS = YPOS - 32.
C
C        CALL MET PROFILE, SUBROUTINE RMETP, TO DETERMINE LAYER VALUE
C
   24 CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RMETP)
      CALL RWDIS(NAME,0)
      CALL GRAF(1)
      CALL CLEAR
      YPOS = 474.
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,50,0,0)
      CALL CODE
      WRITE (IDATAF,100) TOPSUR
      CALL CHAR(400.,YPOS,0,IDATAF,7,0,0)
      ALT(JTOP) = TOPSUR
      YPOS = YPOS - 32.
      GO TO 17
C
C        WRITE OUT THE TOP OF THE SURFACE LAYER AND WIND DIRECTION
C        AND SPEED AT THE TOP
C
   26 CONTINUE
      WRITE (6,205) TOPSUR,IDIR(JTOP),SPEED(JTOP)
C
C        CALCULATE SOURCE STRENGTH
C
      SIGHCL = 2.276E3 * PHCL * QC1 * AA * (TEMP(1) + 273.15)/
               PRESS(1) * TOPSUR**BB
C
C        CALCULATE AND WRITE OUT THE MEAN WIND SPEED, ASPD, AND
C        DIRECTION, ADIR
C
      DO 28 I=2,JTOP
      IF(IABS(IDIR(I) - IDIR(I - 1)) .LT. 180)GO TO 28
      DO 27 J=1,JTOP
   27 IF(IDIR(J) .LT. 180)IDIR(J) = IDIR(J) + 360
      GO TO 31
   28 CONTINUE
```

```
C
   31 ASPD = 0.0
      ADIR = 0.0
      DO 32 I=2,JTOP
      IM1 = I - 1
      DALT = ALT(I) - ALT(IM1)
      ASPD = ASPD + 0.5 * (SPEED(I) + SPEED(IM1)) * DALT
   32 ADIR = ADIR + 0.5 * FLOAT(IDIR(I) + IDIR(IM1)) * DALT
C
      DO 34 I=1,JTOP
   34 IF(IDIR(1) .GT. 360)IDIR(I) = IDIR(I) - 360
C
      DALT = ALT(JTOP) - ALT(1)
      ASPD = ASPD/DALT
      ADIR = ADIR/DALT
      IF(ADIR .GT. 180.0)GO TO 35
      ADIR = ADIR + 180.0
      GO TO 36
   35 ADIR = ADIR - 180.0
C
   36 WRITE (6,206) ASPD,ADIR
C
C         IS THIS A RESEARCH OR A PRODUCTION RUN?
C
      IF(IOPT(2) .EQ. 0)GO TO 45
C
C         RESEARCH RUN -- READ IN SIGA
C
C
C** CALL SUBROUTINE RSIGA TO CALCULATE SIGMA VALUE
C
      J1 = 1
      J2 = 0
      J3 = 0
      DO 41 JJ=1,31
      IF(ABS(ALT(JJ)-304.8).LE.1.0) J3 = JJ
      IF(ABS(PRESS(JJ)-1000.).LE.1.0) J2 = JJ
41    CONTINUE
      IF(J2.EQ.0 .OR.J3.EQ.0) SIGA = 7.0
      IF(J2.EQ.0 .OR.J3.EQ.0) GO TO 42
      CALL RSIGA(J1,J2,J3,RSIG)
      SIGA = RSIG
42    CALL DREAD(NAMEF,6,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      CALL CODE
      WRITE(IDATAF,102) SIGA
      CALL CHAR(330.,YPOS,0,IDATAF,4,0,0)
      CALL IN(2,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      YPOS = YPOS - 32.0
      IF(IX.LE.25) GO TO 45
      CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,62,0,0)
      NIN = 2
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,358.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,101) ISIGA
      IF(NIN.EQ.1)ISIGA = ISIGA/10
      SIGA = FLOAT(ISIGA)
      YPOS = YPOS - 32.
C         WRITE OUT SIGA, THE SIGMA OF THE WIND AZIMUTH ANGLE
C
   45 WRITE (6,208) SIGA
C
      SIGAP = 0.0087266 * SIGA
C
C         CALCULATE THE HORIZONTAL AND VERTICAL CLOUD DIMENSIONS,
C         i.e. SIGXO AND GSPEED
C
      SIGXO = 0.297674 * ALT(31)
      GSPEED = 0.232558 * ALT(31)
C
C         CALCULATE AND WRITE OUT THE EFFECTIVE CLOUD HEIGHT, CLDHT
C
      CLDHT = ALT(31)
      CLDRAD = 2.15 * SIGXO
      IV2 = 0
      IF(CLDRAD+ALT(31).GE.ALT(JTOP))IV2 = 1
      SIGZO = SIGXO
      IF(IV2.EQ.1)SIGZO = (ALT(JTOP) - ALT(31) + CLDRAD)/4.3
      IF(SIGZO.GT.0.0)GO TO 47
      CLDHT = 0.5 * ALT(JTOP)
      SIGZO = 0.64 * CLDHT/2.15
      GO TO 49
   47 IF(IV2.EQ.1)CLDHT = 0.5 * (ALT(JTOP) + ALT(31) - CLDRAD)
C
   49 WRITE (6,209) CLDHT
C
C         CALL THE SEGMENT RCONC
C
      CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RCONC)
      CALL RWDIS(NAME,0)
C
C         END OF RCLDR
C
      END
      SUBROUTINE RWDIS(NAME,JJ)
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER ODCB(144),OBUF(669)
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),ALT(1))
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,669)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,669)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
      DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32),IPAR(5)
      CALL RMPAR(IPAR)
      LU = IPAR(1)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
10    CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ (IBUF,100) (ILINE(I),I=1,32)
100   FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
10    IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS)
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS.LE.48) YPOS = 458.0
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.,YPOS-16.,0,IERS,64,0,0)
      RETURN
      END
      SUBROUTINE RSIGA(J1,J2,J3,RSIG)
C
C*** THIS SUBROUTINE CALCULATES A SIGMA VALUE GIVEN
C*** ALTITUDE, SPEED, TEMP, AND PRESSURE FOR THE
C*** FIRST LEVEL OF DATA, THE 1000FT LEVEL OF DATA
C*** AND THE 1000MB LEVEL OF DATA
C
C
C**            COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
C
      DATA C1,C2,C3,C4,C5,C6/-.008,-.00175,.0008,.50864522,.1132,
     1 3.8163/
      DATA C7/.029/
C CALCULATION OF SIGAR
C NEWTONS METHOD FOR SOLUTION OF F(X,B,D) = 0
      F(X,B,D) =(1.-X**4)/(16.*X**2*(ALOG(D)+C4-2.*ALOG(1.+X)
     1 - ALOG(1.+X**2)+2.*ATAN(X))**2) - B
      FP(X,D) =(-X**4-1.)/(8.*X**3*(ALOG(D)+C4-2.*ALOG(1.+X)
     1 - ALOG(1.+X**2)+2.*ATAN(X))**2) + (1.-X**4)/(2.*(1.+X)
     1 *(1.+X**2)*(ALOG(D)+C4-2.*ALOG(1.+X)-ALOG(1.+X**2)+
     1 2.*ATAN(X))**3)
C
C*** READ IST DATA LEVEL
C
      Z1 = ALT(J1)
      V1 = SPEED(J1)
      T1 = TEMP(J1)
      PZ1 = PRESS(J1)
C
C*** READ 1000MB DATA LEVEL
```

```
C
      Z2 = ALT(J2)
      V2 = SPEED(J2)
      T2 = TEMP(J2)
      PZ2 = PRESS(J2)
C
C*** READ 1000FT DATA LEVEL
C
      Z3 = ALT(J3)
      V3 = SPEED(J3)
      T3 = TEMP(J3)
      PZ3 = PRESS(J3)
C
C   **  CONVERT TO PROPER UNITS
C
      V1 = V1*.514791
      V2 = V2*.514791
      V3 = V3*.514791
      Z1 = Z1*.3048
      Z2 = Z2*.3048
      Z3 = Z3*.3048
      T1 = T1+273.16
      T2 = T2+273.16
      T3 = T3+273.16
C
C*** INITIALIZE Z0
C
      Z0 = .20
C PZ1 AND PZ3 IN MILLIBARS
C V1,V2 AND V3 IN METER/SEC
C Z1,Z2 AND Z3 IN METERS
C T1,T2 AND T3 IN DEG K
C Z0 IN METERS
      E = 22.9183118
      V=V2
      T=(T1+T2+T3)/3.
      Z=(Z1*Z2*Z3)**.33333
      THETA1 = T1*((1000./PZ1)**.288)
      THETA2 = T2
      THETA3 = T3*((1000./PZ3)**.288)
      ZA = (Z1+Z2+Z3)/3.
      THETAA = (THETA1 + THETA2 + THETA3)/3.
      D = Z/Z0
      ZOZ0 = ALOG(D)
      DZTHET = ((Z1-ZA)*(THETA1-THETAA)+(Z2-ZA)*(THETA2-THETAA)
     1          +(Z3-ZA)*(THETA3-THETAA))/((Z1-ZA)**2 + (Z2-ZA)**2
     1          +(Z3-ZA)**2)
      B = 9.8*DZTHET*Z**2/(T*V**2)
      IF(B) 2,25,6
2     CONTINUE
```

```
      R = 1.5
      U = F(R,B,D)
      DO 3 I = 1,50
      R1 = R - F(R,B,D)/FP(R,D)
      U=F(R1,B,D)
      IF(ABS(R1-R).LT.1.E-7) GO TO 21
      IF(I.EQ.49) USAV = U
      IF(I.NE.50) GO TO 888
      IF(USAV.LT.0..AND.U.GT.0..OR.USAV.GT.0..AND.U.LT.0.) GO TO 21
888   CONTINUE
3     R = R1
      RSIG = 30.
      GO TO 1000
6     AP = ZOZO - 1.
      ZOOL10=(C6*ZO)/(7.*Z)
      A1 = 1.
      A2 = 1./(SQRT(B) * 7.*AP)
      A3 = -(AP + 1.)/(7.*AP)
      RAD = A2**2 - 4.*A1*A3
      IF(RAD) 70,80,90
70    CONTINUE
      RSIG = 30.
      GO TO 1000
80    RE11 = -A2/(2.*A1)
      S1 = 1. - 7.*RE11**2
      GO TO 26
90    RE1 = (-A2 + SQRT(RAD))/(2.*A1)
      RI4 = RE1**2
      ZOOL4 = ZO*RI4/(Z*(1. -7.*RI4))
      IF(B.LT.C3) GO TO 37
      IF(B.GE.C3) GO TO 38
21    RI1 = (1.-R1**4)/16.
      ZOOL1 = ZO*RI1/Z
      A = ZOZO +C4-2.*ALOG(1.+R1)-ALOG(1.+R1**2)+2.*ATAN(R1)
      IF(B.LT.C1) GO TO 22
      IF(B.GE.C1.AND.B.LT.C2) GO TO 23
      IF(B.GE.C2) GO TO 24
22    RSIG = E*2.7/A
      GO TO 1000
23    FB2 = 2.7 + 112.*(-C1 + B)
      RSIG = E*FB2/A
      GO TO 1000
24    FB3 = 3.4 - 725.5*(-C2 +B)
      RSIG = E*FB3/A
      GO TO 1000
25    RI2 = 0
      ZOOL2 = 0
      RSIG = 48 816/ALOG(D)
      GO TO 1000
26    RI3 = (S1-1.)/(-7.)
```

```
      Z00L3 = Z0*RI3/(Z*(1. -7.*RI3))
      IF(B.LT.C3) GO TO 27
      IF(B.GE.C3) GO TO 28
27    FB3 = 3.4 - 725.5*(-C2 + B)
      RSIG = (E*FB3)/( 7.*RI3/( 1. -7.*RI3) + Z0Z0 )
      SIGR20=(E*FB3)/(C6+Z0Z0)
      IF(RI3.GE.C5) GO TO 110
      GO TO 1000
110   CONTINUE
      RSIG = SIGR20
      GO TO 1000
28    FB4 = 1.55 + 38.04*(B - .0008)
      RSIG = (E*FB4)/( 7.*RI3/( 1. -7.*RI3) + Z0Z0 )
      SIGR21=(E*FB4)/(C6+Z0Z0)
      IF(RI3.GE.C5) GO TO 115
      GO TO 1000
115   CONTINUE
      RSIG = SIGR21
      GO TO 1000
37    FB3 = 3.4 - 725.5*(-C2+B)
      RSIG = (E*FB3)/( 7.*RI4/(1. - 7.*RI4) + Z0Z0)
      SIGR20=(E*FB3)/(C6+Z0Z0)
      IF(RI4.GE.C5) GO TO 120
      GO TO 1000
120   CONTINUE
      RSIG = SIGR20
      GO TO 1000
38    FB4 = 1.55 + 38.04*(B - .0008)
      FB5 = 2.35 + 5.43*(B - C7)
      RSIG = (E*FB4)/( 7.*RI4/(1. - 7.*RI4) + Z0Z0)
      SIGR21=(E*FB4)/(C6+Z0Z0)
      SIGR22 = (E*FB5)/(C6+Z0Z0)
      IF(RI4.GE.C5.AND.B.LT.C7) GO TO 125
      IF(RI4.GE.C5.AND.B.GE.C7) GO TO 126
      GO TO 1000
125   CONTINUE
      RSIG = SIGR21
      GO TO 1000
126   CONTINUE
      RSIG = SIGR22
      GO TO 1000
C
C*** CHECK FOR VALID SIGA VALUE
C
1000  CONTINUE
      IF (RSIG.LE.0. .OR. RSIG.GT.30.) RSIG = 30.
      RETURN
      END
      END$
```

```
FTN4,L
      PROGRAM RMETP
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION WSX(31),WSY(31),DTX(31),DTY(31),PTX(31),PTY(31),
     1          WDX(31),WDY(31)
      DIMENSION ISTP(3),ITTP(3),ISPT(3),ITPT(3),ISWS(3),ITWS(3)
      DIMENSION ISWD(3),ITWD(3),XDTIC(2),YDTIC(2),ICUR1(21)
      DIMENSION ITEST(10),TPR(6),IDCB(144)
      DIMENSION X(4),Y(4),XTIC1(2),XTIC2(2),YTIC1(2),YTIC2(2)
      DIMENSION XS(2),YS(2),IALTL(8),TSURX(20),BSURX(10)
      DIMENSION IALTCH(336),IALT(22),IHARD(16)
      DIMENSION IXNUM(13),IYNUM(22),IALTC1(8)
      DIMENSION ITEMPD(3),IPRESD(3),IDENSD(3)
      DIMENSION IDATL(2),ITIML(2)
      DIMENSION IDATE(6),AWDIR(31)
      DIMENSION ITMME(2)
      DIMENSION APTEMP(31)
      DIMENSION XL(2),YL(2),IDT(12),IPT(11),IWS(8),IWD(10)
      DIMENSION ISURL1(30),IALTSP(8)
      DIMENSION ISURT(22),ISURT1(16),IALTP(8),IALTCL(8)
      DIMENSION ICRVT(4),ISTL(12),ITOP(2),YWD1(2),YWD2(2),XWD1(18)
      DIMENSION IBOT(2)
      DIMENSION XWD2(2),IWDL1(18),IWDL2(18),IWDL3(18),IWDL4(18)
      DIMENSION ITPV(3),NAME(3)
      DIMENSION IMET(2),INSTAL(2,2),ISTAB(4)
      INTEGER RMETQ(3)
      DATA IHARD/2HHA,2HRD,2H C,2HOP,2HY ,2HDE,2HSI,2HRE,2HD?,2H  ,
      2H  ,2HYE,2HS ,2H  ,2H  ,2HNO/
      DATA RMETQ/2HRM,2HET,1HQ/
      DATA NAME/036522B,2HEE,1HD/
      DATA IWDL1/2H 0,2H  ,2H  ,2H  ,2H 9,2H0 ,2H  ,2H  ,2H18,2H0 ,
                2H  ,2H  ,2H27,2H0 ,2H  ,2H  ,2H36,2H0 /
      DATA IWDL2/2H90,2H  ,2H  ,2H  ,2H18,2H0 ,2H  ,2H  ,2H27,2H0 ,
     .          2H  ,2H  ,2H36,2H0 ,2H  ,2H  ,2H90,2H  /
      DATA IWDL3/2H18,2H0 ,2H  ,2H  ,2H27,2H0 ,2H  ,2H  ,2H36,2H0 ,
     .          2H  ,2H  ,2H90,2H  ,2H  ,2H  ,2H18,2H0 /
      DATA IWDL4/2H27,2H0 ,2H  ,2H  ,2H36,2H0 ,2H  ,2H  ,2H90,2H  ,
     .          2H  ,2H  ,2H18,2H0 ,2H  ,2H  ,2H27,2H0 /
      DATA XWD1/300.,300.,320.,320.,340.,340.,360.,360.,380.,380.,
     .         400.,400.,420.,420.,440.,440.,460.,460./
      DATA ICRVT/2HWS,2HDT,2HPT,2HWD/
      DATA IEXP3/2H3 /
      DATA ISTL/2HSP,2HEE,2HD(,2HM/,2HS),2H  ,2HTE,2HMP,2H(D,2HEG,2H C,
     12H) /
      DATA ICUR1/2HTO,2HUC,2HH ,2HY-,2HAX,2HIS,2H T,2HO ,2HEN,
     12HTE,2HR ,2HTO,2HP ,2HOF,2H S,2HUR,2HFA,2HCE,2H L,2HAY,2HER/
      DATA TPR/139.,187.,236.,285.,334.,383./
      DATA XDTIC/100.,106./
      DATA ITEST/2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  /
      DATA ISURL1/2HSU,2HRF,2HAC,2HE ,2H  ,
     .           2H  ,2HPR,2HES,2HSU,2HRE,
     .           2H  ,2H  ,2H  ,2H  ,2H M,
     .           2HB ,2H  ,2H  ,2H D,2HEN,
     .           2HSI,2HTY,2H  ,2H  ,2H  ,
     .           2H  ,2H G,2H/M,2H  ,2H  /
      DATA IDT/2HDR,2HY ,2HTE,2HMP,2HER,2HAT,2HUR,2HE ,2H(D,2HEG,
     12H C,2H) /
      DATA IPT/2HPO,2HTE,2HNT,2HIA,2HL ,2HTE,2HMP,2H (,2HDE,2HG ,
     12HC)/
      DATA IMINUS/1H-/
      DATA IWS/2HWI,2HND,2H S,2HPE,2HED,2H (,2HM/,2HS)/
      DATA IWD/2HWI,2HND,2H D,2HIR,2HEC,2HTI,2HON,2H (,2HDE,2HG)/
      DATA ISURT/2HSU,2HRF,2HAC,2HE ,2H  ,2H  ,2H  ,2H  ,2HTO,2HP ,
     .          2HLA,2HYE,2HR ,2H  ,2H  ,2H  ,2H  ,2HBO,2HT ,2HLA,
     .          2HYE,2HR /
      DATA ISURT1/2HSU,2HRF,2HAC,2HE ,2H  ,2HTO,2HP ,2HLA,2HYE,2HR ,
     .           2H  ,2HBO,2HT ,2HLA,2HYE,2HR /
      DATA IDATL/2HDA,2HTE/,ITIML/2HTI,2HME/
      DATA IALTL/2H A,2H L,2H T,2H I,2H T,2H U,2H D,2H E/
      DATA TSURX/108.,130.,140.,170.,180.,210.,220.,250.,260.,290.,
     1300.,330.,340.,370.,380.,410.,420.,450.,460.,490./
      DATA BSURX/108.0,160.0,182.5,242.5,265.0,325.0,347.5,407.5,430.0,
     .          490.0/
      DATA ITOP/2H T,2HOP/, IBOT/2H B,2HOT/
      DATA IXNUM/2H10,2H-5,2H 0,2H 5,2H10,2H15,2H20,2H25,2H30,2H35,
     .          2H40,2H45,2H50/
      DATA X/100.,460.,100.,100./
      DATA Y/90.,90.,90.,410./
      DATA IYNUM/2H  ,2H 0,2H 4,2H00,2H 8,2H00,2H12,2H00,2H16,2H00,
     12H20,2H00,2H24,2H00,2H28,2H00,2H32,2H00,2H36,2H00,2H40,2H00/
C ** THIS IS THE ALTERNATE DATA SET WHICH IS BEING CREATED. THESE
C ** CHARACTERS ARE 5 BY 6 RASTER UNITS IN SIZE
```

```
      DATA LCHAR/1H0/,IALT/2H01,2H23,2H45,2H67,2H89,2HAB,2HCD,
     1          2HEF,2HGH,2HIJ,2HKL,2HMN,2HOP,2HQR,2HST,2HUV,
     1          2HWX,2HYZ,2H+-,2H*/,2H()/
C ** THE FOLLOWING DATA STATEMENT CONTAINS OCTAL REPRESENTATION
C ** OF AN ALTERNATE CHARACTER SET AS FOLLOWS: 0-9,A-Z, AND
C ** SPECIAL CHARACTERS +,-,*,/,(,)
      DATA IALTCH/36B,41B,41B,36B,4*0,0,21B,77B,1B,4*0,                 0,1
     1            23B,45B,45B,31B,4*0,42B,41B,51B,66B,4*0,              2,3
     1            14B,24B,77B,4B,4*0,72B,51B,51B,46B,4*0,               4,5
     1            36B,45B,45B,2B,4*0,60B,43B,44B,70B,4*0,               6,7
     1            26B,51B,51B,26B,4*0,20B,51B,51B,36B,4*0,              8,9
     1            37B,50B,50B,37B,4*0,77B,51B,51B,26B,4*0,              A,B
     1            36B,41B,41B,    22B,4*0,77B,41B,41B,36B,4*0,          C,D
     1            77B,51B,51B,41B,4*0,77B,50B,50B,40B,4*0,              E,F
     1            36B,41B,45B,26B,4*0,77B,10B,10B,77B,4*0,              G,H
     1            0,41B,77B,41B,4*0,42B,41B,76B,40B,4*0,                I,J
     1            77B,14B,    22B,41B,4*0,77B,1B,1B,1B,4*0,             K,L
     1            77B,20B,10B,20B,77B,3*0,77B,30B,6B,77B,4*0,           M,N
     1            36B,41B,41B,36B,4*0,77B,44B,44B,30B,4*0,              O,P
     1            34B,42B,42B,35B,4*0,77B,44B,46B,31B,4*0,              Q,R
     1            22B,51B,45B,22B,4*0,40B,40B,77B,40B,40B,3*0,          S,T
     1            76B,1B,1B,76B,4*0,74B,2B,1B,2B,74B,3*0,               U,V
     1            76B,1B,36B,1B,76B,3*0,61B,12B,04B,12B,61B,3*0,        W,X
     1            60B,10B,17B,10B,60B,3*0,41B,43B,45B,51B,61B,3*0,      Y,Z
     1            2*4B,37B,2*4B,3*0,5*4B,3*0,21B,12B,37B,12B,21B,3*0,   +,-
     1            1B,2B,4B,10B,20B,3*0,0,36B,41B,5*0,0,41B,36B,5*0/     /,(
      DATA IALTC1/0,12B,12B,12B,4*0/                                   =,SP
      DATA IALTP/0,1B,6*0/
      DATA IALTCL/0,12B,6*0/
      DATA IALTSP/8*0/
      DATA IMET/2H(M,1H)/
      DATA INSTAL/2HVA,2HFB,2HKS,2HC /
      DATA ISTAB/2HST,2HAB,2H H,2HT:/
C
C*** CALL VERSION SUBROUTINE TO DETERMINE IF RUNNING ON
C*** CRT OR PLASMASCOPE....  IVERSN=0 FOR PLASMA     IVERSN=1 FOR CRT
C
      CALL VERSN(IVERSN)
C ** CALL GRAF(1) TO INITIALIZE PLASMASCOPE
      CALL GRAF(1)
C ** CALL CLEAR TO CLEAR PLASMASCOPE
      CALL CLEAR
C  ** CALL ALTERNATE CHARACTER SET
      CALL LALT(LCHAR,IALTCH,10)
      CALL LALT(1HA,IALTCH(81),26)
      CALL LALT(1H+,IALTCH(289),6)
      CALL LALT(1H=,IALTC1(1),1)
      CALL LALT(1H ,IALTSP,1)
      CALL LALT(1H:,IALTCL,1)
      CALL LALT(1H.,IALTP,1)
```

```
C ** CALL SETOR(XORG,YORG) TO INITIALIZE X,Y ORIGIN
C  ** CALL SETSC(XSCAL,YSCAL) TO SET SCALE FACTORS
      CALL SETSC(1.,1.)
      CALL SETOR(0.,0.)
C  ** READ THE COMMON DISC FILE
C
      CALL RWDIS(NAME,0)
C ** LINE(X,Y,NXY,MODE) TO PLOT LINE
C ** X, Y = CO-ORDINATES
C ** NXY = NUMBER OF POINTS TO BE PLOTTED
C ** MODE = 0 SPECIFIES A WRITE, = 1 SPECIFIES AN ERASE
C ** CALL POINT(X,Y,NXY,MODE) SAME AS ABOVE EXCEFT PLOTS POINTS
C  ** PRINT DATE
      CALL CHAR(20.,490.,0,IDATL,4,2,1)
      XL(1) = 20.
      XL(2) = 48.
      YL(1) = 488.
      YL(2) = 488.
      CALL LINE(XL,YL,2,0)
      CALL CODE
      WRITE(IDATE,3002) ISDAY,ISMON(1),ISMON(2),ISYEAR
3002  FORMAT(I2,1X,A2,A1,1X,I4)
      CALL CHAR(60.,490.,0,IDATE,11,2,1)
C  ** PRINT TIME
      CALL CHAR(164.,490.,0,ITIML,4,2,1)
      XL(1) = 164.
      XL(2) = 192.
      CALL LINE(XL,YL,2,0)
      CALL CODE
      WRITE(ITMME,3001) ISTIM
3001  FORMAT(I4)
      CALL CHAR(204.,490.,0,ITMME,4,2,1)
      CALL CHAR(240.0,490.0,0,IFLAG(4),1,2,1)
      IF(IVERSN.EQ.0)CALL CHAR(248.0,490.0,0,LAUNTD(4),2,2,1)
      IF(IVERSN.EQ.1)CALL CHAR(246.0,490.0,0,LAUNTD(4),2,2,1)
      IF(IFLAG(3).EQ.0)GO TO 2
      I = IFLAG(3) - IFLAG(3)/3
      CALL CHAR(308.0,490.0,0,INSTAL(1,I),4,2,1)
      XL(1) = 308.0
      XL(2) = 336.0
      CALL LINE(XL,YL,2,0)
C  ** PRINT SURFACE PRESSURE AND DENSITY
    2 CALL CHAR(20.,475.,0,ISURL1,60,2,1)
      IF(IVERSN.EQ.0) CALL CHAR(468.,478.,0,IEXP3,1,2,1)
      IF(IVERSN.EQ.1) CALL CHAR(318.,478.,0,IEXP3,1,2,1)
      XL(1) = 20.
      XL(2) = 76.
      YL(1) = 473.
      YL(2) = 473.
      CALL LINE(XL,YL,2,0)
      IF(IVERSN .EQ. 0)GO TO 3
      CALL CHAR(374.0,475.0,0,ISTAB,8,2,1)
      CALL CHAR(466.0,475.0,0,ISTL(4),1,2,1)
      XL(1) = 374.0
      XL(2) = 422.0
      CALL LINE(XL,YL,2,0)
      CALL CODE
      WRITE (IPRESD,2007) ALT(31)
      CALL CHAR(428.0,475.0,0,IPRESD,6,2,1)
C  ** PRINT SURFACE -- TOP LAYER HEADER -- BOT LAYER HEADER (IF REQD)
    3 IF(IVERSN .NE. 0)GO TO 4
      I = 20
      IF(IFLAG(2) .EQ. 1)I = 32
      CALL CHAR(222.0,461.0,0,ISURT1,I,2,1)
      GO TO 5
    4 I = 26
      IF(IFLAG(2) .EQ. 1)I = 44
      CALL CHAR(222.0,461.0,0,ISURT,I,2,1)
    5 XL(1) = 222.
      XL(2) = 278.
      YL(1) = 459.
      YL(2) = 459.
      CALL LINE(XL,YL,2,0)
      XL(1) = 302.
      XL(2) = 374.
      CALL LINE(XL,YL,2,0)
      IF(IFLAG(2) .NE. 1)GO TO 8
      XL(1) = 398.0
      XL(2) = 470.0
      CALL LINE(XL,YL,2,0)
C  ** PRINT DRY TEMPERATURE
    8 CALL CHAR(30.,450.,0,IDT,24,2,1)
C  ** PRINT POTENTIAL TEMPERATURE
      CALL CHAR(30.,440.,0,IPT,22,2,1)
C  ** PRINT WIND SPEED
      CALL CHAR(30.,430.,0,IWS,16,2,1)
C  ** PRINT WIND DIRECTION
      CALL CHAR(30.,420.,0,IWD,20,2,1)
C ** DRAW X AXIS
      CALL LINE(X,Y,2,0)
C ** DRAW Y AXIS
      CALL LINE(X(3),Y(3),2,0)
C ** DO LOOP TO ADD TIC MARKS FOR X AXIS
      XTIC = 70.
      XTIC2(1) = 88.
      XTIC2(2) = 92.
      XNUM1 = 62.
      DO 10 I = 1,13
      XTIC = XTIC + 30.
      XTIC1(1) = XTIC
      XTIC1(2) = XTIC
      CALL LINE(XTIC1,XTIC2,2,0)
      XTIC1(1) = XTIC1(1) + 15.
      XTIC1(2) = XTIC1(2) + 15.
      IF(I.EQ.13) GO TO 13
      CALL LINE(XTIC1,XTIC2,2,0)
13    CONTINUE
      XNUM1 = XNUM1 + 30.
      IF(I.EQ.1) CALL CHAR(84.,80.,0,IMINUS,1,2,1)
      CALL CHAR(XNUM1,80.,0,IXNUM(I),2,2,1)
10    CONTINUE
C   ** DRAW TIC MARKS FOR WIND DIRECTION SCALE
      XWD2(1) = 300.
      XWD2(2) = 460.
      YWD2(1) = 70.
      YWD2(2) = 70.
      YWD1(1) = 68.
      YWD1(2) = 72.
      CALL LINE(XWD2,YWD2,2,0)
      CALL CHAR(310.,50.,0,IWD,20,2,1)
C   ** PRINT LABELS FOR X-AXIS
      CALL CHAR(100.,70.,0,ISTL,24,2,1)
C   ** DO LOOP TO ADD TIC MARKS TO Y-AXIS
      YTIC = 58.
      XTIC2(1) = 98.
      XTIC2(2) = 102.
      DO 20 I = 1,11
      YTIC = YTIC + 32.
      YTIC2(1) = YTIC
      YTIC2(2) = YTIC
      N = (I-1)*2 + 1
      CALL CHAR(64.,YTIC2,0,IYNUM(N),4,2,1)
      CALL LINE(XTIC2,YTIC2,2,0)
20    CONTINUE
C   ** PRINT LABEL FOR Y-AXIS
      YX = 360.
      DO 30 I = 1,8
      YX = YX - 20.
      CALL CHAR(30.,YX,0,IALTL(I),2,2,1)
30    CONTINUE
      CALL CHAR(30.0,YX-20.0,0,IMET,3,2,1)
C   ** THIS PRINTS SURFACE PRESSURE AND DENSITY VALUES
      A = PRESS(1)
      CALL CODE
      WRITE(IPRESD,2007) A
2007  FORMAT(F6.1)
      IF(IVERSN .EQ. 0)CALL CHAR(196.,475.,0,IPRESD,6,2,1)
      IF(IVERSN .EQ. 1)CALL CHAR(133.,475.,0,IPRESD,6,2,1)
      A = SURDEN
      CALL CODE
      WRITE(IDENSD,2007) A
      IF(IVERSN .EQ. 0)CALL CHAR(388.,475.,0,IDENSD,6,2,1)
      IF(IVERSN .EQ. 1)CALL CHAR(260.,475.,0,IDENSD,6,2,1)
C  ** PRINT DRY TEMPERATURES
      A = TEMP(1)
      CALL CODE
      WRITE(ISTP,2007) A
      CALL CHAR(230.,450.,0,ISTP,6,2,1)
C  ** PRINT POTENTIAL TEMPERATURES
      A = PTEMP(1) - 273.15
      CALL CODE
      WRITE(ISPT,2007) A
      CALL CHAR(230.,440.,0,ISPT,6,2,1)
      DO 133 JJ=1,NUM
      IF(ALT(JJ).GE.4000.) GO TO 3131
      WSY(JJ) =( ALT(JJ))*.08+ 90.
      DTY(JJ) =( ALT(JJ))*.08+ 90.
      PTY(JJ) =( ALT(JJ))*.08+ 90.
      WDY(JJ) =( ALT(JJ))*.08+ 90.
      AWDIR(JJ) = IDIR(JJ)
      APTEMP(JJ) = PTEMP(JJ) - 273.15
133   CONTINUE
      JJ = NUM + 1
3131  ILP = JJ - 1
C
C** CALL SUBROUTINE TO ROTATE WIND DIRECTION FOR PLOTTING
C
      CALL WINDS(AWDIR,ILP,ISC)
      DO 123 IK=1,9
      N = (IK-1)*2 + 1
      CALL LINE(XWD1(N),YWD1,2,0)
      XBWD = XWD1(N) - 8.
      YBWD = 60.
      IF(ISC.EQ.1) CALL CHAR(XBWD,YBWD,0,IWDL1(N),4,2,1)
      IF(ISC.EQ.2) CALL CHAR(XBWD,YBWD,0,IWDL2(N),4,2,1)
      IF(ISC.EQ.3) CALL CHAR(XBWD,YBWD,0,IWDL3(N),4,2,1)
      IF(ISC.EQ.4) CALL CHAR(XBWD,YBWD,0,IWDL4(N),4,2,1)
123   CONTINUE
      DO 134 KK=1,ILP
      WSX(KK) =(SPEED(KK))*6. + 160.
      DTX(KK) =(TEMP(KK))*6. + 160.
      PTX(KK) =(APTEMP(KK))*6. + 160.
      IF(TEMP(KK) .LT.-10.)DTX(KK) = 100.
      IF(TEMP(KK) .GT. 50.) DTX(KK) = 460.
      IF(APTEMP(KK) .LT.-10.)PTX(KK) = 100.
      IF(APTEMP(KK) .GT. 50.) PTX(KK) = 460.
      WDX(KK) = ABS(AWDIR(KK))*.444444 + 300.
134   CONTINUE
C  **  PRINT WIND SPEEDS
      A = SPEED(1)
      CALL CODE
      WRITE(ISWS,2007) A
      CALL CHAR(230.,430.,0,ISWS,6,2,1)
C   **  PRINT WIND DIRECTIONS
      A = IDIR(1)
      CALL CODE
      WRITE(ISWD,2007) A
      CALL CHAR(230.,420.,0,ISWD,6,2,1)
C ** THIS PORTION DRAWS THE WIND SPEED LINE
      CALL DLINE(WSX,WSY,ILP,0,8,4)
      XHT = WSY(ILP) + 3.
      CALL CHAR(WSX(ILP),XHT,0,ICRVT(1),2,2,1)
C   ** THIS PORTION DRAWS THE DRY TEMPERATURE LINE
      CALL LINE(DTX,DTY,ILP,0)
      XHT = DTY(ILP) - 5.0
      CALL CHAR(DTX(ILP)+4.0,XHT,0,ICRVT(2),2,2,1)
C **  THIS PORTION DRAWS THE POTENTIAL TEMPERATURE LINE
      CALL DLINE(PTX,PTY,ILP,0,4,4)
      XHT = PTY(ILP) + 3.
      CALL CHAR(PTX(ILP),XHT,0,ICRVT(3),2,2,1)
C **  THIS PORTION DRAWS THE WIND DIRECTION LINE
      I1 = 1
      DO 777 I=2,ILP
      IF(AWDIR(I) .GE. 0.) GO TO 777
      NUMP = I - I1
      CALL DLINE(WDX(I1),WDY(I1),NUMP,0,4,8)
      I1 = I
777   CONTINUE
      NUMP = ILP - I1 + 1
      CALL DLINE(WDX(I1),WDY(I1),NUMP,0,4,8)
      XHT = WDY(ILP) - 5.0
      CALL CHAR(WDX(ILP)+4.0,XHT,0,ICRVT(4),2,2,1)
C   ** THIS PORTION DRAWS TIC MARKS AT VALID DATA POINT OF Y AXIS
      DO 330 K=1,ILP
      YDTIC(1)=  ALT(K)* .08 + 90.
      YDTIC(2) = YDTIC(1)
      CALL LINE(XDTIC,YDTIC,2,0)
330   CONTINUE
C
C         DRAW THE CLOUD
C
      YCLOUD = ALT(31) * 0.08 + 90.0
      CALL CLOUD(250.0,YCLOUD)
C
C         WRITE OUT THE TOP OF THE SURFACE LAYER LINE AND ALLOW IT
C         TO BE MOVED UP AND DOWN
C
      CALL MOVEM(JTOP,ILP,2,ITOP,318.0,TSURX,10)
      TOPSUR = ALT(JTOP)
C
```

```
C          IF REQUESTED, WRITE OUT THE BOTTOM OF THE SURFACE LAYER
C          LINE AND ALLOW IT TO BE MOVED UP AND DOWN
C
      IF(IFLAG(2) .NE. 1)GO TO 444
      CALL MOVEM(JBOT,ILP,1,IBOT,414.0,BSURX,5)
      ZB = ALT(JBOT)
C
C*** CHECK FOR CRT OR PLASMA VERSION
C
  444 IF(IVERSN .EQ. 1)GO TO 446
      CALL CHAR(24.,16.,0,IHARD(1),18,3,0)
      CALL CHAR(168.,16.,0,IHARD(10),8,0,0)
      CALL CHAR(232.,16.,0,IHARD(14),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      IF(IX.GT.15) GO TO 446
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RMETQ)
      CALL RWDIS(NAME,0)
446   CONTINUE
C   ** CALL RWDIS TO PASS CHANGES IN COMMON DIS FILE
C
      CALL RWDIS(NAME,1)
C   ** CALL NGRAF TO RE-INITIATE PLASMASCOPE
      CALL CLEAR
      CALL NGRAF
      STOP
      END
      SUBROUTINE WINDS(WD,NWD,ISC)
      DIMENSION WD(1),ENDPT(4),NUMUP(4)
      EQUIVALENCE (J,LEAST)
      DATA ENDPT/0.0,90.0,180.0,270.0/
      DO 2 I=1,4
    2 NUMUP(I) = 0
      WD2 = WD(1)
      DO 8 I=2,NWD
      WD1 = WD2
      WD2 = WD(I)
      DO 6 J=1,4
      C1 = WD1 - ENDPT(J)
      IF(C1 .LT. 0.0)C1 = C1 + 360.0
      C2 = WD2 - ENDPT(J)
      IF(C2 .LT. 0.0)C2 = C2 + 360.0
      IF(ABS(C1-C2) .LE. 180.0)GO TO 6
      NUMUP(J) = NUMUP(J) + 1
    6 CONTINUE
    8 CONTINUE
      ISC = 1
      LEAST = NUMUP(1)
      DO 12 I=2,4
      IF(NUMUP(I) .GE. LEAST)GO TO 12
      ISC = I
      LEAST = NUMUP(I)
   12 CONTINUE
      DO 17 I=1,NWD
      WD(I) = WD(I) - ENDPT(ISC)
      IF(WD(I) .LT. 0.0)WD(I) = WD(I) + 360.0
   17 CONTINUE
      WD2 = WD(1)
      DO 22 I=2,NWD
      WD1 = WD2
      WD2 = WD(I)
      IF(ABS(WD1-WD2) .LE. 180.0)GO TO 22
      WD(I) = - WD(I)
   22 CONTINUE
      RETURN
      END
      SUBROUTINE CLOUD(XP,YP)
C
C          COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,S8,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION X(181),Y(181)
      RADIUS = GAMMAX * ALT(31) * 0.08
      DO 7 I=1,181
      X(I) = RADIUS * COS(0.01745329252 * FLOAT(2 * I)) + XP
    7 Y(I) = RADIUS * SIN(0.01745329252 * FLOAT(2 * I)) + YP
      CALL LINE(X,Y,181,0)
      RADIUS = 5.0
      X(1) = XP + RADIUS
      X(2) = XP
      X(3) = XP - RADIUS
      X(4) = XP
      X(5) = X(1)
      Y(1) = YP
      Y(2) = YP + RADIUS
      Y(3) = YP
      Y(4) = YP - RADIUS
      Y(5) = Y(1)
      CALL LINE(X,Y,5,0)
      X(2) = XP - RADIUS
      Y(2) = YP
      CALL LINE(X,Y,2,0)
      X(3) = XP
      Y(3) = YP + RADIUS
      CALL LINE(X(3),Y(3),2,0)
      RETURN
      END
      SUBROUTINE MOVEM(JND,MAXJND,MINJND,LAB,XLABEL,XLINE,NLINE)
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
 2000 FORMAT (F6.1)
 2001 FORMAT (" "I3".0")
      INTEGER QUES(13),ANS1,ANS2(2),ANS3(4),BLANKS(26)
      DIMENSION LAB(1),XLINE(1),YLINE(2),JNDALT(3),JNDVAR(3,4)
      EQUIVALENCE (JNDVR1,JNDVAR(1,1)),(JNDVR2,JNDVAR(1,2)),
                  (JNDVR3,JNDVAR(1,3)),(JNDVR4,JNDVAR(1,4))
      DATA QUES/2HMO,2HVE,2H  ,2H  ,2H O,2HF ,2HSU,2HRF,2HAC,2HE ,2HLA,
                2HYE,2HR:/
      DATA ANS1/2HUP/, ANS2/2HDO,2HWN/, ANS3/2HCO,2HNT,2HIN,2HUE/
      DATA BLANKS/26*2H  /
      NEWJND = 0
    1 YLINE(1) = ALT(JND) * 0.08 + 90.0
      YLINE(2) = YLINE(1)
      DO 4 I=1,NLINE
      J = 2 * I - 1
    4 CALL LINE(XLINE(J),YLINE,2,0)
      Y = YLINE(1) + 2.0
      CALL CHAR(460.0,Y,0,LAB,4,2,1)
      Y = Y - 10.0
      CALL CODE
      WRITE (JNDALT,2000) ALT(JND)
      CALL CHAR(460.0,Y,0,JNDALT,6,2,1)
      CALL CODE
      WRITE (JNDVR1,2000) TEMP(JND)
      YLABEL = PTEMP(JND) - 273.15
      CALL CODE
      WRITE (JNDVR2,2000) YLABEL
      CALL CODE
      WRITE (JNDVR3,2000) SPEED(JND)
      CALL CODE
      WRITE (JNDVR4,2001) IDIR(JND)
      YLABEL = 450.0
      DO 6 I=1,4
      CALL CHAR(XLABEL,YLABEL,0,JNDVAR(1,I),6,2,1)
    6 YLABEL = YLABEL - 10.0
      IF(NEWJND .EQ. JND)GO TO 11
      QUES(3) = LAB(1)
      QUES(4) = LAB(2)
      CALL CHARC(0.0,1.0,-1,QUES,26,3,0)
      CALL CHARC(29.0,1.0,-1,ANS1,2,0,0)
      CALL CHARC(36.0,1.0,-1,ANS2,4,0,0)
      CALL CHARC(43.0,1.0,-1,ANS3,8,3,0)
   11 CALL IN(1,J,0.0,0.0,0,0,0,0,31,0,31,I,J)
      IF(I .LE. 20)GO TO 15
      CALL CHAR(0.0,1.0,-1,BLANKS,51,0,0)
      RETURN
   15 IF(I .GE. 17)GO TO 18
      NEWJND = MIN0(JND + 1,MAXJND)
      GO TO 22
   18 NEWJND = MAX0(JND - 1,MINJND)
   22 IF(NEWJND .EQ. JND)GO TO 11
      DO 24 I=1,NLINE
      J = 2 * I - 1
   24 CALL LINE(XLINE(J),YLINE,2,1)
      CALL CHAR(460.0,Y,0,JNDALT,6,1,1)
      Y = Y + 10.0
      CALL CHAR(460.0,Y,0,LAB,4,1,1)
      YLABEL = 450.0
      DO 26 I=1,4
      CALL CHAR(XLABEL,YLABEL,0,JNDVAR(1,I),6,1,1)
   26 YLABEL = YLABEL - 10.0
      JND = NEWJND
      GO TO 1
      END
      SUBROUTINE RWDIS(NAME,JJ)
C
C         COMMON BLOCK
C
```

```
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGNCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER ODCB(144),OBUF(669)
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),ALT(1))
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,669)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,669)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      END$
```

```
FTN4,L
      PROGRAM RMETQ
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,PATO ^,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)`,(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION WSX(31),WSY(31),DTX(31),DTY(31),PTX(31),PTY(31),
     1          WDX(31),WDY(31)
      DIMENSION ISTP(3),ITTP(3),ISPT(3),ITPT(3),ISWS(3),ITWS(3)
      DIMENSION ISWD(3),ITWD(3),XDTIC(2),YDTIC(2),ICUR1(21)
      DIMENSION ITEST(10),TPR(6),IDCB(144)
      DIMENSION X(4),Y(4),XTIC1(2),XTIC2(2),YTIC1(2),YTIC2(2)
      DIMENSION XS(2),YS(2),IALTL(8),TSURX(20),BSURX(10)
      DIMENSION IALTCH(336),IALT(22)
      DIMENSION IXNUM(13),IYNUM(22),IALTC1(8)
      DIMENSION ITEMPD(3),IPRESD(3),IDENSD(3)
      DIMENSION IDATL(2),ITIML(2)
      DIMENSION IDATE(6),AWDIR(31)
      DIMENSION ITMME(2)
      DIMENSION APTEMP(31)
      DIMENSION XL(2),YL(2),IDT(12),IPT(11),IWS(8),IWD(10)
      DIMENSION ISURL1(30),IALTSP(8)
      DIMENSION ISURT(22),IALTP(8),IALTCL(8)
      DIMENSION ICRVT(4),ISTL(12),ITOP(2),YWD1(2),YWD2(2),XWD1(18)
      DIMENSION IBOT(2)
      DIMENSION XWD2(2),IWDL1(18),IWDL2(18),IWDL3(18),IWDL4(18)
      DIMENSION ITPV(3),NAME(3)
      DIMENSION IMET(2),INSTAL(2,2),ISTAB(4)
      DATA NAME/036522B,2HEE,1HD/
      DATA IWDL1/2H 0,2H  ,2H  ,2H  ,2H 9,2H0 ,2H  ,2H  ,2H18,2H0 ,
     .          2H  ,2H  ,2H27,2H0 ,2H  ,2H  ,2H36,2H0 /
      DATA IWDL2/2H90,2H  ,2H  ,2H  ,2H18,2H0 ,2H  ,2H  ,2H27,2H0 ,
     .          2H  ,2H  ,2H36,2H0 ,2H  ,2H  ,2H90,2H  /
      DATA IWDL3/2H18,2H0 ,2H  ,2H  ,2H27,2H0 ,2H  ,2H  ,2H36,2H0 ,
     .          2H  ,2H  ,2H90,2H  ,2H  ,2H  ,2H18,2H0 /
```

```
      DATA IWDL4/2H27,2H0 ,2H  ,2H  ,2H36,2H0 ,2H  ,2H  ,2H90,2H  ,
                2H  ,2H  ,2H18,2H0 ,2H  ,2H  ,2H27,2H0 /
      DATA XWD1/300.,300.,320.,320.,340.,340.,360.,360.,380.,380.,
     1          400.,400.,420.,420.,440.,440.,460.,460./
      DATA ICRVT/2HWS,2HDT,2HPT,2HWD/
      DATA IEXP3/2H3 /
      DATA ISTL/2HSP,2HEE,2HD(,2HM/,2HS),2H  ,2HTE,2HMP,2H(D,2HEG,2H C,
     12H) /
      DATA ICUR1/2HTO,2HUC,2HH ,2HY-,2HAX,2HIS,2H T,2HO ,2HEN,
     12HTE,2HR ,2HTO,2HP ,2HOF,2H S,2HUR,2HFA,2HCE,2H L,2HAY,2HER/
      DATA TPR/139.,187.,236.,285.,334.,383./
      DATA XDTIC/100.,106./
      DATA ITEST/2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  /
      DATA ISURL1/2HSU,2HRF,2HAC,2HE:,2H  ,
     .           2H  ,2HPR,2HES,2HSU,2HRE,
     .           2H  ,2H  ,2H  ,2H  ,2H M,
     .           2HB ,2H  ,2H  ,2H D,2HEN,
     .           2HSI,2HTY,2H  ,2H  ,2H  ,
     .           2H  ,2H G,2H/M,2H  ,2H  /
      DATA IDT/2HDR,2HY ,2HTE,2HMP,2HER,2HAT,2HUR,2HE ,2H(D,2HEG,
     12H C,2H) /
      DATA IPT/2HPO,2HTE,2HNT,2HIA,2HL ,2HTE,2HMP,2H (,2HDE,2HG ,
     12HC)/
      DATA IWS/2HWI,2HND,2H S,2HPE,2HED,2H (,2HM/,2HS)/
      DATA IWD/2HWI,2HND,2H D,2HIR,2HEC,2HTI,2HON,2H (,2HDE,2HG)/
      DATA ISURT/2HSU,2HRF,2HAC,2HE ,2H  ,2H  ,2H  ,2H  ,2HTO,2HP ,
     .          2HLA,2HYE,2HR ,2H  ,2H  ,2H  ,2H  ,2HBO,2HT ,2HLA,
     .          2HYE,2HR /
      DATA IDA.L/2HDA,2HTE/,ITIML/2HTI,2HME/
      DATA IALTL/2H A,2H L,2H T,2H I,2H T,2H U,2H D,2H E/
      DATA IMINUS/1H-/
      DATA TSURX/108.,130.,140.,170.,180.,210.,220.,250.,260.,290.,
     1300.,330.,340.,370.,380.,410.,420.,450.,460.,490./
      DATA BSURX/108.0,160.0,182.5,242.5,265.0,325.0,347.5,407.5,430.0,
     .          490.0/
      DATA ITOP/2H T,2HOP/, IBOT/2H B,2HOT/
      DATA IXNUM/2H10,2H-5,2H 0,2H 5,2H10,2H15,2H20,2H25,2H30,2H35,
     .          2H40,2H45,2H50
      DATA X/100.,460.,100.,100.
      DATA Y/90.,90.,90.,410./
      DATA IYNUM/2H  ,2H 0,2H 4,2H00,2H 8,2H00,2H12,2H00,2H16,2H00,
     12H20,2H00,2H24,2H00,2H28,2H00,2H32,2H00,2H36,2H00,2H40,2H00/
C ** THIS IS THE ALTERNATE DATA SET WHICH IS BEING CREATED, THESE
C ** CHARACTERS ARE 5 BY 6 RASTER UNITS IN SIZE
      DATA LCHAR/1H0/,IALT/2H01,2H23,2H45,2H67,2H89,2HAB,2HCD,
     1          2HEF,2HGH,2HIJ,2HKL,2HMN,2HOP,2HQR,2HST,2HUV,
     1          2HWX,2HYZ,2H+-,2H*/,2H()/
C ** THE FOLLOWING DATA STATEMENT CONTAINS OCTAL REPRESENTATION
C ** OF AN ALTERNATE CHARACTER SET AS FOLLOWS: 0-9,A-Z, AND
C ** SPECIAL CHARACTERS +,-,*,/,(,)
```

```
      DATA IALTCH/36B,41B,41B,36B,4*0,0,21B,77B,1B,4*0,                    0,1
     1            23B,45B,45B,31B,4*0,42B,41B,51B,66B,4*0,                 2,3
     1            14B,24B,77B,4B,4*0,72B,51B,51B,46B,4*0,                  4,5
     1            36B,45B,45B,2B,4*0,60B,43B,44B,70B,4*0,                  6,7
     1            26B,51B,51B,26B,4*0,20B,51B,51B,36B,4*0,                 8,9
     1            37B,50B,50B,37B,4*0,77B,51B,51B,26B,4*0,                 A,B
     1            36B,41B,41B,     22B,4*0,77B,41B,41B,36B,4*0,            C,D
     1            77B,51B,51B,41B,4*0,77B,50B,50B,40B,4*0,                 E,F
     1            36B,41B,45B,26B,4*0,77B,10B,10B,77B,4*0,                 G,H
     1            0,41B,77B,41B,4*0,42B,41B,76B,40B,4*0,                   I,J
     1            77B,14B,     22B,41B,4*0,77B,1B,1B,1B,4*0,               K,L
     1            77B,20B,10B,20B,77B,3*0,77B,30B,6B,77B,4*0,              M,N
     1            36B,41B,41B,36B,4*0,77B,44B,44B,30B,4*0,                 O,P
     1            34B,42B,42B,35B,4*0,77B,44B,46B,31B,4*0,                 Q,R
     1            22B,51B,45B,22B,4*0,40B,40B,77B,40B,40B,3*0,             S,T
     1            76B,1B,1B,76B,4*0,74B,2B,1B,2B,74B,3*0,                  U,V
     1            76B,1B,36B,1B,76B,3*0,61B,12B,04B,12B,61B,3*0,           W,X
     1            60B,10B,17B,10B,60B,3*0,41B,43B,45B,51B,61B,3*0,         Y,Z
     1            2*4B,37B,2*4B,3*0,5*4B,3*0,21B,12B,37B,12B,21B,3*0,      +,-
     1            1B,2B,4B,10B,20B,3*0,0,36B,41B,5*0,0,41B,36B,5*0/        /,(
      DATA IALTC1/0,12B,12B,12B,4*0/                                       =,SP
      DATA IALTP/0,1B,6*0/
      DATA IALTCL/0,12B,6*0/
      DATA IALTSP/8*0/
      DATA IMET/2H(M,1H)/
      DATA INSTAL/2HVA,2HFB,2HKS,2HC /
      DATA ISTAB/2HST,2HAB,2H H,2HT:/
C ** CALL GRAF(1) TO INITIALIZE PLASMASCOPE
      CALL GRAF(1)
C ** CALL CLEAR TO CLEAR PLASMASCOPE
C     CALL CLEAR
C  ** CALL ALTERNATE CHARACTER SET
      CALL LALT(LCHAR,IALTCH,10)
      CALL LALT(1HA,IALTCH(81),26)
      CALL LALT(1H+,IALTCH(289),6)
      CALL LALT(1H=,IALTC1(1),1)
      CALL LALT(1H ,IALTSP,1)
      CALL LALT(1H:,IALTCL,1)
      CALL LALT(1H.,IALTP,1)
C ** CALL SETOR(XORG,YORG) TO INITIALIZE X,Y ORIGIN
C  ** CALL SETSC(XSCAL,YSCAL) TO SET SCALE FACTORS
      CALL SETSC(1.,1.)
      CALL SETOR(0.,0.)
C  ** READ THE COMMON DISC FILE
C
      CALL RWDIS(NAME,0)
C ** LINE(X,Y,NXY,MODE) TO PLOT LINE
C ** X, Y = CO-ORDINATES
C ** NXY = NUMBER OF POINTS TO BE PLOTTED
C ** MODE = 0 SPECIFIES A WRITE, = 1 SPECIFIES AN ERASE
```

```
C ** CALL POINT(X,Y,NXY,MODE) SAME AS ABOVE EXCEPT PLOTS POINTS
C   ** PRINT DATE
      CALL CHAR(20.,490.,0,IDATL,4,2,1)
      XL(1) = 20.
      XL(2) = 48.
      YL(1) = 488.
      YL(2) = 488.
      CALL LINE(XL,YL,2,0)
      CALL CODE
      WRITE(IDATE,3002) ISDAY,ISMON(1),ISMON(2),ISYEAR
3002  FORMAT(I2,1X,A2,A1,1X,I4)
      CALL CHAR(60.,490.,0,IDATE,11,2,1)
C   ** PRINT TIME
      CALL CHAR(164.,490.,0,ITIML,4,2,1)
      XL(1) = 164.
      XL(2) = 192.
      CALL LINE(XL,YL,2,0)
      CALL CODE
      WRITE(ITMME,3001) ISTIM
3001  FORMAT(I4)
      CALL CHAR(204.,490.,0,ITMME,4,2,1)
      CALL CHAR(240.0,490.0,0,IFLAG(4),1,2,1)
      CALL CHAR(246.0,490.0,0,LAUNTD(4),2,2,1)
      IF(IFLAG(3) .EQ. 0)GO TO 2
      I = IFLAG(3) - IFLAG(3)/3
      CALL CHAR(308.0,490.0,0,INSTAL(1,I),4,2,1)
      XL(1) = 308.0
      XL(2) = 336.0
      CALL LINE(XL,YL,2,0)
C   ** PRINT SURFACE PRESSURE AND DENSITY
    2 CALL CHAR(20.,475.,0,ISURL1,60,2,1)
      CALL CHAR(318.,478.,0,IEXP3,1,2,1)
      XL(1) = 20.
      XL(2) = 76.
      YL(1) = 473.
      YL(2) = 473.
      CALL LINE(XL,YL,2,0)
      CALL CHAR(374.0,475.0,0,ISTAB,8,2,1)
      CALL CHAR(466.0,475.0,0,ISTL(4),1,2,1)
      XL(1) = 374.0
      XL(2) = 422.0
      CALL LINE(XL,YL,2,0)
      CALL CODE
      WRITE (IPRESD,2007) ALT(31)
      CALL CHAR(428.0,475.0,0,IPRESD,6,2,1)
C   ** PRINT SURFACE -- TOP LAYRT HEADER -- BOT LAYER HEADER (IF REQD)
      I = 26
      IF(IFLAG(2) .EQ. 1)I = 44
      CALL CHAR(222.0,461.0,0,ISURT,I,2,1)
      XL(1) = 222.
      XL(2) = 278.
      YL(1) = 459.
      YL(2) = 459.
      CALL LINE(XL,YL,2,0)
      XL(1) = 302.
      XL(2) = 374.
      CALL LINE(XL,YL,2,0)
      IF(IFLAG(2).NE.1) GO TO 8
      XL(1) = 398.
      XL(2) = 470.
      CALL LINE(XL,YL,2,0)
C   ** PRINT DRY TEMPERATURE
8     CALL CHAR(30.,450.,0,IDT,24,2,1)
C   ** PRINT POTENTIAL TEMPERATURE
      CALL CHAR(30.,440.,0,IPT,22,2,1)
C   ** PRINT WIND SPEED
      CALL CHAR(30.,430.,0,IWS,16,2,1)
C   ** PRINT WIND DIRECTION
      CALL CHAR(30.,420.,0,IWD,20,2,1)
C ** DRAW X AXIS
      CALL LINE(X,Y,2,0)
C ** DRAW Y AXIS
      CALL LINE(X(3),Y(3),2,0)
C ** DO LOOP TO ADD TIC MARKS FOR X AXIS
      XTIC = 70.
      XTIC2(1) = 88.
      XTIC2(2) = 92.
      XNUM1 = 62.
      DO 10 I = 1,13
      XTIC = XTIC + 30.
      XTIC1(1) = XTIC
      XTIC1(2) = XTIC
      CALL LINE(XTIC1,XTIC2,2,0)
      XTIC1(1) = XTIC1(1) + 15.
      XTIC1(2) = XTIC1(2) + 15.
      IF(I.EQ.13) GO TO 13
      CALL LINE(XTIC1,XTIC2,2,0)
13    CONTINUE
      XNUM1 = XNUM1 + 30.
      IF(I.EQ.1) CALL CHAR(84.,80.,0,IMINUS,1,2,1)
      CALL CHAR(XNUM1,80.,0,IXNUM(I),2,2,1)
10    CONTINUE
C   ** DRAW TIC MARKS FOR WIND DIRECTION SCALE
      XWD2(1) = 300.
      XWD2(2) = 460.
      YWD2(1) = 70.
      YWD2(2) = 70.
      YWD1(1) = 68.
      YWD1(2) = 72.
      CALL LINE(XWD2,YWD2,2,0)
      CALL CHAR(310.,50.,0,IWD,20,2,1)
C  ** PRINT LABELS FOR X-AXIS
      CALL CHAR(100.,70.,0,ISTL,24,2,1)
C  ** DO LOOP TO ADD TIC MARKS TO Y-AXIS
      YTIC = 58.
      XTIC2(1) = 98.
      XTIC2(2) = 102.
      DO 20 I = 1,11
      YTIC = YTIC + 32.
      YTIC2(1) = YTIC
      YTIC2(2) = YTIC
      N = (I-1)*2 + 1
      CALL CHAR(64.,YTIC2,0,IYNUM(N),4,2,1)
      CALL LINE(XTIC2,YTIC2,2,0)
20    CONTINUE
C  ** PRINT LABEL FOR Y-AXIS
      YX = 360.
      DO 30 I = 1,8
      YX = YX - 20.
      CALL CHAR(30.,YX,0,IALTL(I),2,2,1)
30    CONTINUE
      CALL CHAR(30.,YX-20.,0,IMET,3,2,1)
C  ** THIS PRINTS SURFACE PRESSURE AND DENSITY VALUES
      A = PRESS(1)
      CALL CODE
      WRITE(IPRESD,2007) A
2007  FORMAT(F6.1)
      CALL CHAR(133.,475.,0,IPRESD,6,2,1)
      A = SURDEN
      CALL CODE
      WRITE(IDENSD,2007) A
      CALL CHAR(260.,475.,0,IDENSD,6,2,1)
C  ** PRINT DRY TEMPERATURES
      A = TEMP(1)
      CALL CODE
      WRITE(ISTP,2007) A
      CALL CHAR(230.,450.,0,ISTP,6,2,1)
C  ** PRINT POTENTIAL TEMPERATURES
      A = PTEMP(1) - 273.!5
      CALL CODE
      WRITE(ISPT,2007) A
      CALL CHAR(230.,440.,0,ISPT,6,2,1)
      DO 133 JJ=1,NUM
      IF(ALT(JJ).GE.4000.) GO TO 3131
      WSY(JJ) =( ALT(JJ))*.08+ 90.
      DTY(JJ) =( ALT(JJ))*.08+ 90.
      PTY(JJ) =( ALT(JJ))*.08+ 90.
      WDY(JJ) =( ALT(JJ))*.08+ 90.
      AWDIR(JJ) = IDIR(JJ)
      APTEMP(JJ) = PTEMP(JJ) - 273.15
```

```
133    CONTINUE
       JJ = NUM + 1
3131   ILP = JJ - 1
C
C** CALL SUBROUTINE TO ROTATE WIND DIRECTION FOR PLOTTING
C
       CALL WINDS(AWDIR,ILP,ISC)
       DO 123 IK=1,9
       N = (IK-1)*2 + 1
       CALL LINE(XWD1(N),YWD1,2,0)
       XBWD = XWD1(N) - 8.
       YBWD = 60.
       IF(ISC.EQ.1) CALL CHAR(XBWD,YBWD,0,IWDL1(N),4,2,1)
       IF(ISC.EQ.2) CALL CHAR(XBWD,YBWD,0,IWDL2(N),4,2,1)
       IF(ISC.EQ.3) CALL CHAR(XBWD,YBWD,0,IWDL3(N),4,2,1)
       IF(ISC.EQ.4) CALL CHAR(XBWD,YBWD,0,IWDL4(N),4,2,1)
123    CONTINUE
       DO 134 KK=1,ILP
       WSX(KK) =(SPEED(KK))*6. + 160.
       DTX(KK) =(TEMP(KK))*6. + 160.
       PTX(KK) =(APTEMP(.K))*6. + 160.
       IF(TEMP(KK) .LT.-10.)DTX(KK) = 100.
       IF(TEMP(KK) .GT. 50.) DTX(KK) = 460.
       IF(APTEMP(KK) .LT.-10.)PTX(KK) = 100.
       IF(APTEMP(KK) .GT. 50.) PTX(KK) = 460.
       WDX(KK) = ABS(AWDIR(KK))*.44444 + 300.
134    CONTINUE
C   **  PRINT WIND SPEEDS
       A = SPEED(1)
       CALL CODE
       WRITE(ISWS,2007) A
       CALL CHAR(230.,430.,0,ISWS,6,2,1)
C   **  PRINT WIND DIRECTIONS
       A = IDIR(1)
       CALL CODE
       WRITE(ISVD,2007) A
       CALL CHAR(230.,420.,0,ISWD,6,2,1)
C ** THIS PORTION DRAWS THE WIND SPEED LINE
       CALL DLINE(WSX,WSY,ILP,0,8,4)
       XHT = WSY(ILP) + 3.
       CALL CHAR(WSX(ILP),XHT,0,ICRVT(1),2,2,1)
C   ** THIS PORTION DRAWS THE DRY TEMPERATURE LINE
       CALL LINE(DTX,DTY,ILP,0)
       XHT = DTY(ILP) - 5.0
       CALL CHAR(DTX(ILP)+4.0,XHT,0,ICRVT(2),2,2,1)
C **   THIS PORTIUN DRAWS THE POTENTIAL TEMPERATURE LINE
       CALL DLINE(PTX,PTY,ILP,0,4,4)
       XHT = PTY(ILP) + 3.
       CALL CHAR(PTX(ILP),XHT,0,ICRVT(3),2,2,1)
C **   THIS PORTION DRAWS THE WIND DIRECTION LINE
```

```
      I1 = 1
      DO 777 I=2,ILP
      IF(AWDIR(I) .GE. 0.) GO TO 777
      NUMP = I - I1
      CALL DLINE(WDX(I1),WDY(I1),NUMP,0,4,8)
      I1 = I
777   CONTINUE
      NUMP = ILP - I1 + 1
      CALL DLINE(WDX(I1),WDY(I1),NUMP,0,4,8)
      XHT = WDY(ILP) - 5.0
      CALL CHAR(WDX(ILP)+4.0,XHT,0,ICRVT(4),2,2,1)
C  ** THIS PORTION DRAWS TIC MARKS AT VALID DATA POINT OF Y AXIS
      DO 330 K=1,ILP
      YDTIC(1)=  ALT(K)* .08 + 90.
      YDTIC(2) = YDTIC(1)
      CALL LINE(XDTIC,YDTIC,2,0)
330   CONTINUE
C
C          DRAW THE CLOUD
C
      YCLOUD = ALT(31) * 0.08 + 90.0
      CALL CLOUD(250.0,YCLOUD)
C
C          WRITE OUT THE TOP OF THE SURFACE LAYER LINE
C
      CALL MOVEM(JTOP,ILP,2,ITOP,318.0,TSURX,10)
C
C          IF REQUESTED, WRITE OUT THE BOTTOM OF THE SURFACE LAYER LINE
C
      IF(IFLAG(2) .NE. 1)GO TO 444
      CALL MOVEM(JBOT,ILP,1,IBOT,414.0,BSURX,5)
C
C          CALL NGRAF TO REINITIALIZE PLASMASCOPE
C
C     CALL CLEAR
444   CONTINUE
      CALL NGRAF
      STOP
      END
      SUBROUTINE WINDS(WD,NWD,ISC)
      DIMENSION WD(1),ENDPT(4),NUMUP(4)
      EQUIVALENCE (J,LEAST)
      DATA ENDPT/0.0,90.0,180.0,270.0/
      DO 2 I=1,4
    2 NUMUP(I) = 0
      WD2 = WD(1)
      DO 8 I=2,NWD
      WD1 = WD2
      WD2 = WD(I)
      DO 6 J=1,4
      C1 = WD1 - ENDPT(J)
      IF(C1 .LT. 0.0)C1 = C1 + 360.0
      C2 = WD2 - ENDPT(J)
      IF(C2 .LT. 0.0)C2 = C2 + 360.0
      IF(ABS(C1-C2) .LE. 180.0)GO TO 6
      NUMUP(J) = NUMUP(J) + 1
    6 CONTINUE
    8 CONTINUE
      ISC = 1
      LEAST = NUMUP(1)
      DO 12 I=2,4
      IF(NUMUP(I) .GE. LEAST)GO TO 12
      ISC = I
      LEAST = NUMUP(I)
   12 CONTINUE
      DO 17 I=1,NWD
      WD(I) = WD(I) - ENDPT(ISC)
      IF(WD(I) .LT. 0.0)WD(I) = WD(I) + 360.0
   17 CONTINUE
      WD2 = WD(1)
      DO 22 I=2,NWD
      WD1 = WD2
      WD2 = WD(I)
      IF(ABS(WD1-WD2) .LE. 180.0)GO TO 22
      WD(I) = - WD(I)
   22 CONTINUE
      RETURN
      END
      SUBROUTINE CLOUD(XP,YP)
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .        IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .        ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .        LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .        SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .        SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .        TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .        YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
 2000 FORMAT (F6.1)
      DIMENSION X(181),Y(181)
      RADIUS = GAMMAX * ALT(31) * 0.08
      DO 7 I=1,181
      X(I) = RADIUS * COS(0.01745329252 * FLOAT(2 * I)) + XP
    7 Y(I) = RADIUS * SIN(0.01745329252 * FLOAT(2 * I)) + YP
      CALL LINE(X,Y,181,0)
      RADIUS = 5.0
      X(1) = XP + RADIUS
      X(2) = XP
      X(3) = XP - RADIUS
      X(4) = XP
      X(5) = X(1)
      Y(1) = YP
      Y(2) = YP + RADIUS
      Y(3) = YP
      Y(4) = YP - RADIUS
      Y(5) = Y(1)
      CALL LINE(X,Y,5,0)
      X(2) = XP - RADIUS
      Y(2) = YP
      CALL LINE(X,Y,2,0)
      X(3) = XP
      Y(3) = YP + RADIUS
      CALL LINE(X(3),Y(3),2,0)
      RETURN
      END
      SUBROUTINE MOVEM(JND,MAXJND,MINJND,LAB,XLABEL,XLINE,NLINE)
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
 2000 FORMAT (F6.1)
 2001 FORMAT (" "I3".0")
      INTEGER QUES(13),ANS1,ANS2(2),ANS3(4),BLANKS(26)
      DIMENSION LAB(1),XLINE(1),YLINE(2),JNDALT(3),JNDVAR(3,4)
      EQUIVALENCE (JNDVR1,JNDVAR(1,1)),(JNDVR2,JNDVAR(1,2)),
               (JNDVR3,JNDVAR(1,3)),(JNDVR4,JNDVAR(1,4))
      DATA QUES/2HMO,2HVE,2H  ,2H  ,2H O,2HF ,2HSU,2HRF,2HAC,2HE ,2HLA,
              2HYE,2HR:/
      DATA ANS1/2HUP/, ANS2/2HDO,2HWN/, ANS3/2HCO,2HNT,2HIN,2HUE/
      DATA BLANKS/26*2H  /
      NEWJND = 0
    1 YLINE(1) = ALT(JND) * 0.08 + 90.0
      YLINE(2) = YLINE(1)
      DO 4 I=1,NLINE
      J = 2 * I - 1
    4 CALL LINE(XLINE(J),YLINE,2,0)
      Y = YLINE(1) + 2.0
      CALL CHAR(460.0,Y,0,LAB,4,2,1)
      Y = Y - 10.0
      CALL CODE
      WRITE (JNDALT,2000) ALT(JND)
      CALL CHAR(460.0,Y,0,JNDALT,6,2,1)
      CALL CODE
      WRITE (JNDVR1,2000) TEMP(JND)
      YLABEL = PTEMP(JND) - 273.15
      CALL CODE
      WRITE (JNDVR2,2000) YLABEL
      CALL CODE
      WRITE (JNDVR3,2000) SPEED(JND)
      CALL CODE
      WRITE (JNDVR4,2001) IDIR(JND)
      YLABEL = 450.0
      DO 6 I=1,4
      CALL CHAR(XLABEL,YLABEL,0,JNDVAR(1,I),6,2,1)
    6 YLABEL = YLABEL - 10.0
      RETURN
      END
      SUBROUTINE RWDIS(NAME,JJ)
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER ODCB(144),OBUF(669)
```

```
DIMENSION NAME(3)
EQUIVALENCE (OBUF(1),ALT(1))
CALL OPEN(ODCB,IERR,NAME,0)
IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,669)
IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,669)
CALL CLOSE(ODCB,IERR)
RETURN
END
END$
```

```
FTN4,L
      PROGRAM RCONC
C
C*****************************************************************
C
C          CONCENTRATION AND DOSAGE PROGRAM -- A PROGRAM OF THE
C          REED SERIES OF PROGRAMS
C
C*****************************************************************
C
C
C          COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C          OUTPUT FORMAT STATEMENTS
C
  200 FORMAT ("1"12X"CLOUD CONCENTRATIONS AND DOSAGES"/
             "0DISTANCE"4X"CONCENTRATION"5X"DOSAGE"6X
              "TIME AFTER LAUNCH(SEC)"/
              " (METERS)"8X"(PPM)"8X"(PPM SEC)"8X"START"3X"FINISH")
  201 FORMAT (1XF7.1,8XF7.3,8XF7.3,9XF5.1,3XF5.1)
  202 FORMAT (//"0****POINT OF MAXIMUM CONCENTRATION****"/
              6X"RANGE FROM PAD(M): "F8.1/
              6X"DIRECTION(DEG): "F5.1/
              6X"HEIGHT(M): "F6.1/
              6X"MAXIMUM CONCENTRATION(PPM): "F6.3)
  203 FORMAT (//"0****CONCENTRATIONS AND DOSAGES WITH 10 DEGREE "
              "UNCERTAINTIES****")
  204 FORMAT ("0"5X"RANGE(M): "F7.1/
              6X"AZIMUTH(DEG): "F5.1/
              6X"MATERIAL"5X"CONCENTRATION(PPM)"11X"DOSAGE(PPM)")
  205 FORMAT (4I5,I2)
  206 FORMAT (7X3A2,6XF8.3" +/- "F8.3,4XF8.3" +/- "F8.3)
C
```

```
C          TYPE AND DIMENSION STATEMENTS
C
      LOGICAL IGRAF
      INTEGER RISOP(3)
      DIMENSION FACT(3),CMNPL(3),DMNPL(3),MATS(3,5),NAME(3),
     *          NAMEF(3),ILINE(32),IDATAF(10),IERS(32),
     *          DISTV(81),DOSV(81),CONCV(81)
C
C          DATA STATEMENTS
      DATA NAME/036522B,2HEE,1HD/,NAMEF/2H?R,2HCO,2HNC/
      DATA IERS/32*2H  /
C
      DATA FACT/0.0,-0.174533,0.174533/
      DATA MATS/2H  ,2HMC,2HL ,2H  ,2H C,2HO ,
     *          2H  ,2HCO,2H2 ,2H A,2HL2,2HO3,
     *          2H  ,2H N,2HO /
      DATA RISOP/2HRI,2HSO,1HP/
C
C          CALL GRAF TO INITIALIZE SCOPE (APPROPRIATE ONLY WHEN USING
C          PLASMASCOPE)
C
      CALL GRAF(1)
C
C          READ COMMON DISK FILE #REEDD
C
      CALL RWDIS(NAME,0)
C
C          IF THIS IS A RESEARCH RUN, DETERMINE IF PLOTTING IS DESIRED
C
      IF(IOPT(2) .EQ. 0)GO TO 55
C
      CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,42,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(IX .LE. 25)IGRAF = .TRUE.
      IF(IX .GT. 25)IGRAF = .FALSE.
C
C          DO LOOP FOR CONCENTRATION AND DOSAGE CALCULATIONS
C
C          DIST - RANGE FROM STABILIZATION
C          DOSPK - DOSAGE
C          DOSMAX - MAXIMUM DOSAGE
C          CONCPK - CONCENTRATION
```

```
C          CONMAX - MAXIMUM CONCENTRATION
C
   55 NUMV = 0
      CONMAX = 0.0
      DOSMAX = 0.0
      ACTVOL = PI43 * CLDRAD * CLDRAD * CLDRAD
      TOTVOL = ACTVOL
      IF(IV2 .EQ. 1)ACTVOL = PI * (ALT(JTOP) + CLDRAD - ALT(31))**2 *
                           (2.0 * CLDRAD - ALT(JTOP) + ALT(31))/3.0
      SIGHCL = SIGHCL * ACTVOL/TOTVOL
C
      WRITE (6,200)
C
      DO 59 I=0,20000,250
C
      NUMV = NUMV + 1
      DIST = I
      DISTV(NUMV) = DIST
C
      CALL DFEXP(JTOP,1000.0)
C
      DOSPK = SIGHCL * E1/(TWOPI * R2 * ASPD * SQRT(0.5 * R3))
      DOSV(NUMV) = DOSPK
      CONCPK = DOSPK * ASPD/(SQR2PI * SIGX)
      CONCV(NUMV) = CONCPK
C
      DOSMAX = AMAX1(DOSPK,DOSMAX)
C
      IF(CONCPK .LE. CONMAX)GO TO 58
      RATOMC = DIST
      CONMAX = CONCPK
      SGXMAX = SIGX
      SGYMAX = SIGY
C
   58 IF(AMOD(DIST,1000.0) .NE. 0.0)GO TO 59
C
      ARG1 = CRTIME(31) + (DIST - AL1)/ASPD
      ARG2 = CRTIME(31) + (DIST + AL1)/ASPD
      WRITE (6,201) DIST,CONCPK,DOSPK,ARG1,ARG2
C
   59 CONTINUE
C
C          IF REQUESTED, PLOT THE CENTERLINE DOSAGE AND CONCENTRATION
C          VALUES
C
      ARG1 = ALOGT(DOSMAX)
      IEXPD = ARG1
      IF(ARG1 .LT. 0.0)IEXPD = IEXPD - 1
      IEXPD = - IEXPD
      ARG1 = ALOGT(CONMAX)
      IEXPC = ARG1
      IF(ARG1 .LT. 0.0)IEXPC = IEXPC - 1
      IEXPC = - IEXPC
      IF(.NOT. IGRAF)GO TO 61
      CALL CPLOT(DISTV,DOSV,CONCV,NUMV,IEXPD,IEXPC)
C
C        CALCULATE AND WRITE OUT THE POINT OF MAXIMUM CONCENTRATION
C
   61 ARG1 = DEGRAD * ADIR
      DIST = RATOMC * COS(ARG1)
      Y1 = RATOMC * SIN(ARG1)
C
      DO 62 I=2,JTOP
      IF(CLDHT .LE. ALT(I))GO TO 63
   62 CONTINUE
      I = JTOP
C
   63 IM1 = I - 1
      RANGSR = SAVER(IM1) + (CLDHT - ALT(IM1)) *
                (SAVER(I) - SAVER(IM1))/(ALT(I) - ALT(IM1))
C
      ARG1 = SAVEA(I) - SAVEA(IM1)
      IF(ABS(ARG1) .LT. 180.0)GO TO 66
      IF(ARG1 .GT. 0.0)GO TO 65
      SAVEA(I) = SAVEA(I) + 360.0
      GO TO 66
   65 SAVEA(IM1) = SAVEA(IM1) + 360.0
C
   66 AZCS = SAVEA(IM1) + (CLDHT - ALT(IM1)) * (SAVEA(I) - SAVEA(IM1))/
                           (ALT(I) - ALT(IM1))
      IF(AZCS .GE. 360.0)AZCS = AZCS - 360.0
C
      ARG1 = DEGRAD * AZCS
      X2 = RANGSR * COS(ARG1)
      Y2 = RANGSR * SIN(ARG1)
      X = DIST + X2
      Y = Y1 + Y2
C
      RNGE = SQRT(X * X + Y * Y)
      DIR = RADDEG * ATAN2(Y,X)
      IF(DIR .LT. 0.0)DIR = DIR + 360.0
      WRITE (6,202) RNGE,DIR,ZB,CONMAX
C
C        IF THIS IS A PRODUCTION RUN, SKIP THE OFF CENTER CONCENTATION
C        SECTION AND THE CALL OF PROGRAM RISOP -- IF PLOTTING WAS NOT
C        REQUESTED, JUST SKIP THE OFF CENTER CONCENTRATION SECTION
C
      IF(IGRAF)GO TO 68
      IF(IOPT(2) .EQ. 0)GO TO 88
      GO TO 81
```

```
C
C          OFF CENTER CONCENTRATIONS SECTION
C
   68 CALL LABEL(IEXPD,IEXPC)
C
C          ARE OFF CENTER CONCENTRATIONS DESIRED?
C
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,38,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,0,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,3,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX.LE.25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX.GT.25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(IX.GT.25)GO TO 81
C
C          OFF CENTER CONCENTRATIONS ARE DESIRED
C
      WRITE (6,203)
C
      CALL ORGIN(IXSET,IYSET)
C
      ARG1 = 0.0
      IF(ADIR.GT.180.0)ARG1 = 360.0
      BETAF = DEGRAD * (180.0 + ARG1 - ADIR)
C
      ARG1 = 0.0
      IF(AZCS.GT.180.0)ARG1 = 360.0
      BETAS = DEGRAD * (180.0 + ARG1 - AZCS)
      XP = RANGSR * COS(BETAS)
      YP = RANGSR * SIN(BETAS)
C
      ITER = 0
C
C          LOOP ON OFF CENTER CONCENTRATION REQUESTS
C
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
   71 ITER = ITER + 1
C
C          READ IN AND WRITE OUT THE RANGE AND AZIMUTH FOR THE
C          OFF CENTER CONCENTRATION CALCULATION -- ENTERING A RANGE OF 0
C          TERMINATES THE PROCEDURE
C
      IF(YPOS.LT.48.) YPOS = 458.
      CALL DREAD(NAMEF,6,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      NIN = 7
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,112.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) RP
      IF(RP .LE. 0.0)GO TO 78
      NIN = 7
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,272.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) AZP
      YPOS = YPOS - 16.
      IF(YPOS .LT. 48.0)YPOS = 458.0
      WRITE (6,204) RP,AZP
C
      ARG1 = 0.0
      IF(AZP .GT. 180.0)ARG1 = 360.0
      AP = DEGRAD * (180.0 + ARG1 - AZP)
      XS = RP * COS(AP)
      YS = RP * SIN(AP)
C
C         ON THE PLOTTER, WRITE OUT AN ASTERISK AND THE ITERATION
C         NUMBER AT THE LOCATION WHERE THE OFF CENTER CONCENTRATION
C         CALCULATION IS DESIRED
C
      IX = IXSET + 0.2631 * XS
      IY = IYSET + 0.3545 * YS
      WRITE (12) -1,1,IX,IY
      CALL SYMBL(100,125,1H*)
      IX = IX + 75
      WRITE (12) -1,1,IX,IY
      WRITE (12,205) 100,0,0,125,ITER
C
C         CALCULATE THE CONCENTRATIONS AND DOSAGES AT THIS POINT PLUS
C         10 DEGREES UNCERTAINTIES ON EITHER SIDE
C
      XHAT = XS - XP
      YHAT = YS - YP
C
      DO 74 I=1,3
      ARG1 = BETAF - FACT(I)
      Y = - XHAT * SIN(ARG1) + YHAT * COS(ARG1)
      DIST = XHAT * COS(ARG1) + YHAT * SIN(ARG1)
      CALL DFEXP(JTOP,1000.0)
      DOS = SIGHCL * E1 * EXP(- Y * Y/(2.0 * R2 * R2))/
     1      (TWOPI * R2 * ASPD * SQRT(0.5 * R3))
      CONC = DOS * ASPD/(SQR2PI * SIGX)
      CMNPL(I) = CONC
   74 DMNPL(I) = DOS
C
C        CALCULATE AND WRITE OUT THE CONCENTRATION AND DOSAGE FOR
C        EACH MATERIAL
C
      DELC = ABS(0.5 * (2.0 * CMNPL(1) - CMNPL(2) - CMNPL(3)))
      DELD = ABS(0.5 * (2.0 * DMNPL(1) - DMNPL(2) - DMNPL(3)))
      WRITE (6,206) (MATS(I,1),I=1,3),CMNPL(1),DELC,DMNPL(1),DELD
C
      ARG1 = PCO/PHCL
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,206) (MATS(I,2),I=1,3),CONC,DLC,DOS,DLD
C
      ARG1 = PCO2/PHCL
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,206) (MATS(I,3),I=1,3),CONC,DLC,DOS,DLD
C
      ARG1 = PAL2O3/PHCL * 0.43882420 * PRESS(1)/
                                   (TEMP(1) + 273.16)
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,206) (MATS(I,4),I=1,3),CONC,DLC,DOS,DLD
C
      ARG1 = PNO/PHCL
      CONC = ARG1 * CMNPL(1)
      DLC = ARG1 * DELC
      DOS = ARG1 * DMNPL(1)
      DLD = ARG1 * DELD
      WRITE (6,206) (MATS(I,5),I=1,3),CONC,DLC,DOS,DLD
C
C        REQUEST ANOTHER POINT FOR AN OFF CENTER CONCENTRATION
C        CALCULATION
C
      GO TO 71
C
C        OFF CENTER CONCENTRAIONS ARE NOT DESIRED
C
   78 CONTINUE
      YPOS = YPOS - 16.
C
C        IS AN ISOPLETH PLOT DESIRED?
```

```
C
81    CALL DREAD(NAMEF,4,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,26,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
C
C        IF AN ISOPLETH PLOT IS DESIRED, CALL THE PROGRAM RISOP
C
      IF(IX .GE. 28)GO TO 87
      CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RISOP)
      CALL RWDIS(NAME,0)
      CALL GRAF(1)
   87 CONTINUE
C
C        CALL NGRAF TO RETURN SCOPE TO NORMAL MODE OF OPERATION
C        (APPROPRIATE ONLY WHEN USING PLASMA SCOPE)
C
   88 CALL NGRAF
C
C        WRITE OUT THE COMMON DISK FILE
C
      CALL RWDIS(NAME,1)
C
C        RETURN TO THE MAIN PROGRAM REED
C
      STOP
C
C        END OF RCONC
C
      END
      SUBROUTINE RWDIS(NAME,JJ)
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .              (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .              (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .              (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .              (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .              (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER ODCB(144),OBUF(669)
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),ALT(1))
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,669)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,669)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE CPLOT(DISTV,DOSV,CONCV,NUMV,IEXPD,IEXPC)
C
C*******************************************************************
C
C        THIS SUBROUTINE PLOTS THE DOSAGE AND CONCENTRATION CENTERLINE
C        CURVES
C
C*******************************************************************
C
C
C        COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .              (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .              (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .              (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .              (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .              (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C        DIMENSION STATEMENT
C
      DIMENSION DISTV(1),DOSV(1),CONCV(1)
C
C        CALCULATE PLOTTING FACTORS
C
      FDIST = 9295.0/30000.0
      FDOS = 8231.0 * 10.0**(IEXPD - 1)
      FCONC = 8231.0 * 10.0**(IEXPC - 1)
C
C        PLOT THE DOSAGE CENTERLINE CURVE
C
      DO 7 I=1,NUMV
      IX = DISTV(I) * FDIST + 725.0
      IY = DOSV(I) * FDOS + 1040.0
      WRITE (12) -1,1,IX,IY
    7 CALL SYMBL(100,100,25400B)
C
C        PLOT THE CONCENTRATION CENTERLINE CURVE
C
      DO 16 I=1,NUMV
      J = 1/I
      J = 1 - 2 * J
      IX = DISTV(I) * FDIST + 725.0
      IY = CONCV(I) * FCONC + 1040.0
   16 WRITE (12) J,1,IX,IY
C
C        RETURN TO RCONC
C
      RETURN
C
C        END OF CPLOT
C
      END
      SUBROUTINE LABEL(IEXPD,IEXPC)
C
C***********************************************************************
C
C        THIS SUBROUTINE LABELS THE CONCENTRATION AND DOSAGE CENTERLINE
C        PLOTS
C
C***********************************************************************
C
C
C        COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,S8,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .           (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .           (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .           (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .           (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .           (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C        OUTPUT FORMAT STATEMENTS
C
  200 FORMAT (4I5,I2)
  201 FORMAT (4I5,F5.0)
  202 FORMAT (4I5,F5.2)
  203 FORMAT (4I5,I4,1XA1,A2,2XI2,1XA2,A1,1XI4)
  204 FORMAT (4I5,I4' C'A2,2XI2,1XA2,A1,1XI4)
  205 FORMAT (4I5,I4,1XR1,A2,2XI2,1XA2,A1,1XI4)
C
C        LABEL THE PLOT
C
      I = - IEXPC
      WRITE (12) -1,1,300,5000
      WRITE (12,200) 0,150,-100,0,I
      I = - IEXPD
      WRITE (12) -1,1,300,6500
      WRITE (12,200) 0,150,-100,0,I
      WRITE (12) -1,1,3700,8950
      WRITE (12,201) 125,0,0,125,CLDHT
      WRITE (12) -1,1,3700,8745
      WRITE (12,201) 125,0,0,125,CRTIME(31)
      WRITE (12) -1,1,3700,8540
      WRITE (12,202) 125,0,0,125,CONMAX
      WRITE (12) -1,1,3700,8335
      WRITE (12,201) 125,0,0,125,ALT(JTOP)
      WRITE (12) -1,1,3700,8130
      WRITE (12,201) 125,0,0,125,ZB
      WRITE (12) -1,1,3700,7925
      WRITE (12,201) 125,0,0,125,ZB
      IF(IOPT(1) .EQ. 1)GO TO 4
      WRITE (12) -1,1,5625,8980
      WRITE (12) 1,1,6125,8980
      GO TO 7
    4 WRITE (12) -1,1,5025,8980
      WRITE (12) 1,1,5525,8980
      WRITE (12) -1,1,5725,8950
      WRITE (12,203) 125,0,0,125,ISTIM,IFLAG(4),LAUNTD(4),ISDAY,ISMON,
                    ISYEAR
    7 WRITE (12) -1,1,5725,8695
      WRITE (12,204) 125,0,0,125,ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
      WRITE (12) -1,1,5725,8490
      IF(LTIME)WRITE (12,205) 125,0,0,125,LTIM,LAUNTD(3),LAUNTD(4),LDAY,
                            LMON,LYEAR
C
```

```
C          RETURN TO RCONC
C
      RETURN
C
C          END OF LABEL
C
      END
      SUBROUTINE DFEXP(J,CONC)
C
C*******************************************************************
C
C          THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS
C
C          J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER
C          CONC - CONCENTRATION TO BE TESTED
C
C*******************************************************************
C
C
C          COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,S8,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C          CALCULATE SIGMA Z
C
      SIGZ = DIST * SIGAP + SIGZ0/1.28
      R3 = 2.0 * SIGZ * SIGZ
C
C          CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION
C
      TWOI = 2.0
      ZT = ALT(J)
      TEMP2 = CLDHT - ZZ
      TEMP3 = CLDHT - 2.0 * ZB + ZZ
      E1 = EXP( - TEMP2 * TEMP2/R3) +
     1         EXP( - TEMP3 * TEMP3/R3)
    4 TEMP1 = TWOI * (ZT - ZB)
      TEXPSM = E1
      TEXP = (TEMP1 - TEMP2)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP2)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 - TEMP3)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP3)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      IF(E1 .EQ. TEXPSM)GO TO 7
      TWOI = TWOI + 2.0
      GO TO 4
    7 E1 = REFLEC * E1
C
C        CALCULATE SIGMA Y
C
      S8 = DIST * SIGAP + SIGXO
      R2 = SQRT(S8 * S8 + (0.0040589 * FLOAT(IDIR(J) - IDIR(1)) *
     1                      DIST)**2)
C
C        CALCULATE CLOUD LENGTH
C
      TEMP1 = SPEED(J) - SPEED(1)
      AL1 = 0.28 * TEMP1 * DIST/ASPD
      IF(TEMP1 .GE. 0.0)GO TO 11
      IF(PTEMP(J)-PTEMP(1) .GT. 0.0)AL1 = 0.0
      AL1 = ABS(AL1)
C
C        CALCULATE SIGMA X
C
   11 SIGX = SQRT((AL1/4.3)**2 + SIGXO * SIGXO)
C
C        IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN
C        TO THE CALLING PROGRAM
C
      IF(CONC .EQ. 1000.0)RETURN
C
C        CALCULATE CROSS WIND DISTANCE
C
      Y1 = - 2.0 * R2 * R2 * ALOG(15.7496 * CONC * SIGX * R2 *
     1                                   SIGZ/(SIGHCL * E1))
      Y1 = SQRT(AMAX1(Y1,0.0))
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
```

```
C          END OF DFEXP
C
      END
      SUBROUTINE ORGIN(IXO,IYO)
C
C****************************************************************
C
C          THIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING
C          FOR THE COMPLEX AND MAP SELECTED
C
C****************************************************************
C
C
C          COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION ILINE(32),IDATAF(10),IERS(32),IMAPL(48),NAMEF(3)
C
C          INPUT FORMAT STATEMENT
C
  100 FORMAT (I2,1XA1)
C
C          OUTPUT FORMAT STATEMENT
C
C
C          TYPE AND DIMENSION STATEMENTS
C
      LOGICAL NOT1ST
      DIMENSION IX(8),IY(8)
C
C          DATA STATEMENTS
C
      DATA IERS/32*2H  /
      DATA NAMEF/2H?R,2HIS,2HOP/
      DATA IMAPL/2H40,2H,S,2HEA,2H M,2HAP,2H
```

```
     1            2H40,2H.L,2HAN,2HD ,2HMA,2HP ,
     1            2H41,2H.S,2HEA,2H M,2HAP,2H  ,
     1            2H41,2H.L,2HAN,2HD ,2HMA,2HP ,
     1            2H17,2H.S,2HEA,2H M,2HAP,2H  ,
     1            2H17,2H.L,2HAN,2HD ,2HMA,2HP ,
     1            2H39,2H.S,2HEA,2H M,2HAP,2H  ,
     1            2H39,2H.L,2HAN,2HD ,2HMA,2HP /
      DATA NOT1ST/.FALSE./, LCHAR/1HL/
      DATA IX/5450,5411,4830,4825,8750,8730,4100,4100/
      DATA IY/2630,8243,2465,8050,2990,8600,1700,7300/
C
C         IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
C         IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
C         COORDINATES, I, AGAIN
C
      IF(NOT1ST)GO TO 7
C
C         THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
C         AND THE DESIRED MAP, i.e. SEA OR LAND
C
      NOT1ST = .TRUE.
      CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      IF(YPOS.LT.48.) YPOS = 458.
      IF(IOPT(3).EQ.1) CALL DREAD(NAMEF,8,ILINE)
      IF(IOPT(3).EQ.2) CALL DREAD(NAMEF,9,ILINE)
      IF(IOPT(3).EQ.0) CALL DREAD(NAMEF,10,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(24.,YPOS,0,ILINE(2),8,3,0)
      CALL CHAR(95.,YPOS,0,ILINE(7),50,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IXC,IYC)
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(200.,YPOS+16,0,IERS,25,0,0)
      IF(IXC.LT.6.AND.IOPT(3).EQ.1) I=1
      IF(IXC.GT.5.AND.IXC.LT.12.AND.IOPT(3).EQ.1) I=2
      IF(IXC.GT.11.AND.IXC.LT.18.AND.IOPT(3).EQ.1) I=3
      IF(IXC.GT.17.AND.IOPT(3).EQ.1) I=4
      IF(IXC.LT.6.AND.IOPT(3).EQ.2) I=5
      IF(IXC.GE.6.AND.IOPT(3).EQ.2) I=6
      IF(IXC.LT.6.AND.IOPT(3).EQ.0) I=7
      IF(IXC.GE.6.AND.IOPT(3).EQ.0) I=8
      IMP = (I - 1)*6 + 1
      CALL CHAR(208.,YPOS+16.,0,IMAPL(IMP),12,0,0)
      YPOS = YPOS - 16.
      IF(YPOS .LT.48.) YPOS = 458.
C
C         SET THE COORDINATES BASED ON THE INDEX I
C
```

```
    7 IX0 = IX(I)
      IY0 = IY(I)
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF ORGIN
C
      END
      SUBROUTINE SYMBL(IWIDE,IHI,ISYMB)
      IX=-IWIDE/2
      IY=-IHI/2
      WRITE(12) -1,-1,IX,IY
      WRITE(12,100) IWIDE,0,0,IHI,ISYMB
100   FORMAT(4I5,A1,1H_)
      IY=-IY
      WRITE(12)-1,-1,IX,IY
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
      DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32),IPAR(5)
      CALL RMPAR(IPAR)
      LU = IPAR(1)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
10    CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
100   FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
10    IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS)
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS.LE.48) YPOS = 458.0
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
CALL CHAR(0.,YPOS-16.,0,IERS,64,0,0)
RETURN
END
END$
```

```
FTN4,L
      PROGRAM RISOP
C
C**********************************************************************
C
C         ISOPLETH PLOTTING PROGRAM -- A PROGRAM IN THE REED SERIES
C         OF PROGRAMS
C
C**********************************************************************
C
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
             SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
             TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
             YPOS,IFLAG(5),ZP,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
                  (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
                  (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
                  (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
                  (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C         OUTPUT FORMAT STATEMENTS
C
  200 FORMAT ("1"20X"CLOUD LOCATION AND DIMENSIONS"/
             " TIME FROM CLOUD STABILIZATION"5X"RANGE"5X"AZIMUTH"
             8X"DIAMETERS (METERS)"/
             11X"(MINUTES)"14X"(METERS)"4X"(DEG)"6X"CROSS WIND"
             4X"ALONG WIND")
  201 FORMAT (12XF6.2,16XF8.1,4XF5.1,7XF7.1,7XF7.1)
  202 FORMAT (4I5,I4" C"A2,2XI2,1XA2,A1,1XI4)
  203 FORMAT (4I5,A1)
  204 FORMAT (4I5,F5.2"_")
  205 FORMAT (4I5", "F5.2"_")
  206 FORMAT (F5.3)
  207 FORMAT (I1)
  208 FORMAT (4I5,I4,1XR1,A2,2XI2,1XA2,A1,1XI4)
C
C         TYPE AND DIMENSION STATEMENTS
C
      LOGICAL DFALTC
      DIMENSION CONC(10),NAME(3),NAMEF(3),ILINE(32),IDATAF(10),
                IERS(32),IXA(100),IYA(100),IXB(100),IYB(100)
      DATA NAME/036522B,2HEE,1HD/,NAMEF/2H?R,2HIS,2HOP/
      DATA IERS/32*2H  /
C
C        CALL GRAF TO INITIALIZE SCOPE (APPROPRIATE ONLY WHEN USING
C        PLASMASCOPE)
C
      CALL GRAF(1)
C
C        READ COMMON DISK FILE
C
      CALL RWDIS(NAME,0)
C
C        DETERMINE THE ORIGIN ON THE MAP FOR THIS PLOT AND MOVE THE
C        PEN THERE
C
      CALL ORGIN(IXO,IYO)
      WRITE (12) -1,1,IXO,IYO
C
C        DETERMINE THE INDEX IN THE ALTITUDE DATA ARRAY THAT HAS
C        THAT ALTITUDE JUST LOWER THAN THE EFFECTIVE CLOUD HEIGHT, CLDHT
C
      DO 4 I=2,JTOP
      IF(CLDHT .GT. ALT(I))GO TO 4
      ICLDHT = I - 1
      GO TO 5
    4 CONTINUE
      ICLDHT = JTOP
C
C        DRAW THE LINE DEPICTING CLOUD MOVEMENT ALONG THE GROUND
C        AS FAR AS THE CLOUD STABILIZATION POINT
C
    5 X = 0.0
      Y = 0.0
      DO 9 I=2,ICLDHT
      IM1 = I - 1
      RANGE = 0.5 * (CRTIME(I) - CRTIME(IM1)) * (SPEED(I) + SPEED(IM1))
      DIR = 0.5 * FLOAT(IDIR(I) + IDIR(IM1))
      IF(IABS(IDIR(I) - IDIR(IM1)) .GT. 180)DIR = DIR - 180.0
      IF(DIR .LT. 0.0)DIR = DIR + 360.0
      DIR = DEGRAD * (360.0 - DIR)
      X = X + RANGE * COS(DIR)
      Y = Y + RANGE * SIN(DIR)
      IX = INT(0.2631 * X) + IXO
      IY = INT(0.3545 * Y) + IYO
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 11
    9 WRITE (12) 1,1,IX,IY
C
C        MAKE THE CALCULATIONS NECESSARY TO WRITE OUT THE CLOUD
C        LOCATION AND DIMENSIONS
```

```
C
   11 ALT1 = 0.5 * (CLDHT + ALT(ICLDHT))
      ICLDP1 = ICLDHT + 1
      ARG1 = ALT(ICLDP1) - ALT(ICLDHT)
      ARG2 = (CLDHT - ALT(ICLDHT))/ARG1
      SPCENT = SPEED(ICLDHT) + (SPEED(ICLDP1) - SPEED(ICLDHT)) * ARG2
      RANGE = SPCENT * (CRTIME(ICLDP1) - CRTIME(ICLDHT)) * ARG2
      IF(IABS(IDIR(ICLDP1) - IDIR(ICLDHT)) .LT. 180)GO TO 14
      IF(IDIR(ICLDP1) .LT. 180)IDIR(ICLDHT) = IDIR(ICLDHT) + 360
      IF(IDIR(ICLDHT) .LT. 180)IDIR(ICLDHT) = IDIR(ICLDHT) + 360
   14 DIR = FLOAT(IDIR(ICLDHT)) + (ALT1 - ALT(ICLDHT)) *
                                  FLOAT(IDIR(ICLDP1) - IDIR(ICLDHT))/ARG1
      IF(DIR .GT. 360.0)DIR = DIR - 360.0
      IF(DIR .GT. 180.0)GO TO 17
      DIR = DIR + 180.0
      GO TO 18
   17 DIR = DIR - 180.0
   18 DIR = 180.0 - DIR
      ARG1 = DEGRAD * DIR
      X = X + RANGE * COS(ARG1)
      Y = Y + RANGE * SIN(ARG1)
      R = SQRT(X * X + Y * Y)
      DELR = 300.0 * ASPD
C
      DACRS = 4.30 * SIGX0
      DALNG = 4.30 * SIGX0
C
      ARG1 = 180.0
      IF(DIR .GT. 180.0)ARG1 = 540.0
      AZ = ARG1 - DIR
C
      ARG1 = 180.0
      IF(ADIR .GT. 180.0)ARG1 = 540.0
      DAZ = ARG1 - ADIR
      ARG1 = DEGRAD * DAZ
      DELX = DELR * COS(ARG1)
      DELY = DELR * SIN(ARG1)
C
      DELU = ABS(SPEED(ICLDHT) - SPEED(1))
C
      DELTH = IDIR(JTOP) - IDIR(1)
C
      TIM = 0.0
      R1 = 0.0
      XC = X
      YC = Y
      TXL = 0.28 * DELU/ASPD
      SIGX02 = SIGX0 * SIGX0
      S02 = S0 * S0
      WRITE (6,200)
```

```
C
      DO 22 I=1,13
      WRITE (6,201) TIM,R,AZ,DACRS,DALNG
      TIM = TIM + 5.0
      R1 = R1 + DELR
      XL = R1 * TXL
      SIGX = SQRT((XL/4.30)**2 + SIGX02)
      DACRS = 4.30 * SIGX
      SIGY = SQRT(S82 + (0.0040589 - 3.0 * DELTH * R1)**2)
      DALNG = 4.30 * SIGY
      XC = XC + DELX
      YC = YC + DELY
      R = SQRT(XC * XC + YC * YC)
   22 AZ = 180.0 - RADDEG * ATAN2(YC,XC)
C
C         LABEL THE CLOUD STABILIZATION POINT WITH A +
C
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 77
      IXX = IX
      IYY = IY
      WRITE (12) 1,1,IX,IY
      CALL SYMBL(150,150,1H+)
C
C         LABEL THE POINT OF MAXIMUM CONCENTRATION WITH A @
C
      DIR = DEGRAD * (180.0 - ADIR)
      CDIR = COS(DIR)
      SDIR = SIN(DIR)
      IX1 = INT(0.2631 * (X + RATOMC * CDIR)) + IX0
      IY1 = INT(0.3545 * (Y + RATOMC * SDIR)) + IY0
      WRITE (12) -1,1,IX1,IY1
      CALL SYMBL(150,150,1H@)
C
C         DRAW THE LINE OF CLOUD MOVEMENT ALONG THE GROUND FROM
C         THE CLOUD STABILIZATION POINT ON
C
      WRITE (12) -1,1,IXX,IYY
      RANGE = 1000.0
   27 X = X + RANGE * CDIR
      Y = Y + RANGE * SDIR
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 29
      WRITE (12) 1,1,IX,IY
      GO TO 27
   29 WRITE (12) -1,1,IXX,IYY
C
C         ARE DEFAULT CONCENTRATION VALUES GOING TO BE USED
```

```
C          FOR THE PLOTS
C
      IF(YPOS.LT.48.) YPOS = 458.
      CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 32.
C
C          YES -- SET UP THE DEFAULT VALUES
C
      CONC(1) = 0.1 * CONMAX
      CONC(2) = 0.5 * CONMAX
      CONC(3) = 0.75 * CONMAX
      CONC(4) = - 1.0
      CALL CODE
      WRITE (IDATAF,206) CONC(1)
      CALL CHAR(440.,YPOS+48.,0,IDATAF,5,0,0)
      CALL CODE
      WRITE (IDATAF,206) CGNC(2)
      CALL CHAR(120.,YPOS+32.,0,IDATAF,5,0,0)
      CALL CODE
      WRITE (IDATAF,206) CONC(3)
      CALL CHAR(256.,YPOS+32.,0,IDATAF,5,0,0)
      CALL DREAD(NAMEF,4,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,46,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX.LE.25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX.GT.25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(YPOS.LT.64.0)YPOS = 458.0
      DFALTC = .FALSE.
      IF(IX.LT.28)DFALTC = .TRUE.
C
C          DO LOOP OVER THE 10 POSSIBLE CONCENTRATION VALUES FOR THE PLOTS
C
      IF(DFALTC)GO TO 35
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPJS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 32.
      IF(YPOS.LE.64) YPOS=458.
   35 DO 59 I=1,10
```

```
C
C         IF DEFAULT CONCENTRATION VALUES ARE NOT BEING USED,
C         READ IN THE VALUE FOR THIS PLOT
C
      IF(DFALTC)GO TO 37
      CALL DREAD(NAMEF,6,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,17,3,0)
      CALL CODE
      WRITE (IDX,207) I
      CALL CHAR(111.,YPOS,0,IDX,1,3,0)
      NIN = 9
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,144.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) CONC(I)
      CALL CHAR(0.,YPOS,0,ILINE,17,0,0)
      CALL CHAR(111.,YPOS,0,IDX,1,0,0)
      YPOS = YPOS - 16.
      IF(YPOS .LT. 48.0)YPOS = 458.0
   37 IF(CONC(I) .LT. 0.0)GO TO 61
C
C         ITERATE TO FIND THE LOCATION OF THIS CONCENTRATION
C         ON THE PLOT
C
      DIST = 0.0
      DINC = 1000.0
C
   41 CALL DFEXP(JTOP,CONC(I))
      IF(Y1 .GT. 0.0)GO TO 42
      DIST = DIST + DINC
      GO TO 41
C
   42 IF(DINC .LE. 100.0)GO TO 43
      DIST = DIST - 900.0
      DINC = 100.0
      GO TO 41
C
   43 IF(DINC .LE. 10.0)GO TO 44
      DIST = DIST - 90.0
      DINC = 10.0
      GO TO 41
C
C         PLOT OUT THE CONCENTRATION LINE ON BOTH SIDES
C
   44 DIST = DIST - 10.0
      IX = INT(0.2631 * DIST * CDIR) + IXX
      IY = INT(0.3545 * DIST * SDIR) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 59
      NUMA = 1
      IXA(NUMA) = IX
      IYA(NUMA) = IY
      NUMB = 1
      IXB(NUMB) = IX
      IYB(NUMB) = IY
C
      DIST = DIST + 10.0
      IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
      IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 59
      NUMA = 2
      IXA(NUMA) = IX
      IYA(NUMA) = IY
C
      IX = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
      IY = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
      NUMB = 2
      IXB(NUMB) = IX
      IYB(NUMB) = IY
C
   46 DIST = DIST + 500.0
      CALL DFEXP(JTOP,CONC(I))
      IX = INT(0.2631 * (DIST * CDIR - Y1 * SDIR)) + IXX
      IY = INT(0.3545 * (DIST * SDIR + Y1 * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
      NUMA = NUMA + 1
      IXA(NUMA) = IX
      IYA(NUMA) = IY
C
      IF(Y1 .GT. 0.0)GO TO 52
      NUMB = NUMB + 1
      IXB(NUMB) = IX
      IYB(NUMB) = IY
      GO TO 54
C
   52 IX = INT(0.2631 * (DIST * CDIR + Y1 * SDIR)) + IXX
      IY = INT(0.3545 * (DIST * SDIR - Y1 * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
      NUMB = NUMB + 1
      IXB(NUMB) = IX
      IYB(NUMB) = IY
      GO TO 46
C
   54 WRITE (12) -1,1,IXA(1),IYA(1)
      DO 56 J=2,NUMA
   56 WRITE (12) 1,1,IXA(J),IYA(J)
      IF(NUMB .EQ. 1)GO TO 59
      WRITE (12) -1,1,IXB(1),IYB(1)
      DO 57 J=2,NUMB
   57 WRITE (12) 1,1,IXB(J),IYB(J)
C
   59 CONTINUE
C
C        ON THE PLOT, CROSS OUT EITHER THE WORD FORECAST OR SOUNDING
C
   61 IF(IOPT(1) .NE. 0)GO TO 62
      WRITE (12) -1,1,707,604
      WRITE (12) 1,1,1174,604
      GO TO 64
C
   62 WRITE (12) -1,1,1269,604
      WRITE (12) 1,1,1760,604
C
C        PRINT OUT THE PREDICTION TIME ON THE PLOT
C
   64 WRITE (12) -1,1,1869,319
      WRITE (12,202) 100,0,0,150,ITIME,LAUNTD(4),IDAY,MONTH,IYEAR
C
C        IF THE LAUNCH TIME WAS ENTERED, PRINT IT OUT ON THE PLOT
C
      IF(.NOT. LTIME)GO TO 67
      WRITE (12) -1,1,1869,112
      WRITE (12,208) 100,0,0,150,LTIM,LAUNTD(3),LAUNTD(4),LDAY,
                    LMON,LYEAR
C
C        ON THE PLOT, PRINT OUT THE CHARACTERS + AND @ FOR THE LEGEND
C
   67 WRITE (12) -1,1,1041,1342
      WRITE (12,203) 150,0,0,150,1H+
      WRITE (12) -1,1,1041,1104
      WRITE (12,203) 150,0,0,150,1H@
C
C        FOR THE LEGEND ON THE PLOT, PRINT OUT THE CONCENTRATION VALUES
C        FOR WHICH CONTOURS WERE DRAWN
C
      WRITE (12) -1,1,1066,9587
      DO 75 I=1,10
      IF(CONC(I) .LT. 0.0)GO TO 77
      IF(I .NE. 1)GO TO 72
      WRITE (12,204) 125,0,0,150,CONC(I)
      GO TO 75
   72 WRITE (12,205) 125,0,0,150,CONC(I)
   75 CONTINUE
C
C        WRITE OUT COMMON DISK FILE
C
   77 CALL RWDIS(NAME,1)
C
C        CALL NGRAF TO RETURN SCOPE TO NORMAL MODE OF OPERATION
```

```
C           (APPROPRIATE ONLY WHEN USING PLASMASCOPE)
C
      CALL NGRAF
C
C           RETURN TO THE MAIN PROGRAM REED
C
      STOP
C
C           END OF RISOP
C
      END
      SUBROUTINE RWDIS(NAME,JJ)
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     *       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     *       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     *       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     *       SIGMCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     *       SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
     *       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     *       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     *            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     *            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     *            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     *            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     *            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER ODCB(144),OBUF(669)
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),ALT(1))
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,669)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,669)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE DFEXP(J,CONC)
C
C*****************************************************************
C
C           THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS
C
C           J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER
C           CONC - CONCENTRATION TO BE TESTED
C
C*****************************************************************
C
C
C           COMMON BLOCK
```

```
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
     .       IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
     .       ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
     .       LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
     .       SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .       SIGX0,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZ0,SIGAP,SB,TEMP(31),
     .       TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .       YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C        CALCULATE SIGMA Z
C
      SIGZ = DIST * SIGAP + SIGZ0/1.28
      R3 = 2.0 * SIGZ * SIGZ
C
C        CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION
C
      TWOI = 2.0
      ZT = ALT(J)
      TEMP2 = CLDHT - ZZ
      TEMP3 = CLDHT - 2.0 * ZB + ZZ
      E1 = EXP( - TEMP2 * TEMP2/R3) +
     .          EXP( - TEMP3 * TEMP3/R3)
    4 TEMP1 = TWOI * (ZT - ZB)
      TEXPSM = E1
      TEXP = (TEMP1 - TEMP2)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP2)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 - TEMP3)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP3)**2/R3
      IF(TEXP .LE. 120.0)E1 = E1 + EXP( - TEXP)
      IF(E1 .EQ. TEXPSM)GO TO 7
      TWOI = TWOI + 2.0
      GO TO 4
    7 E1 = REFLEC * E1
C
C        CALCULATE SIGMA Y
C
      SB = DIST * SIGAP + SIGX0
      R2 = SQRT(S8 * S8 + (0.0040589 * FLOAT(IDIR(J) - IDIR(1)) *
                                  DIST)**2)
C
C         CALCULATE CLOUD LENGTH
C
      TEMP1 = SPEED(J) - SPEED(1)
      AL1 = 0.28 * TEMP1 * DIST/ASPD
      IF(TEMP1 .GE. 0.0)GO TO 11
      IF(PTEMP(J)-PTEMP(1) .GT. 0.0)AL1 = 0.0
      AL1 = ABS(AL1)
C
C         CALCULATE SIGMA X
C
   11 SIGX = SQRT((AL1/4.3)**2 + SIGX0 * SIGX0)
C
C         IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN
C         TO THE CALLING PROGRAM
C
      IF(CONC .EQ. 1000.0)RETURN
C
C         CALCULATE CROSS WIND DISTANCE
C
      Y1 = - 2.0 * R2 * R2 * ALOG(15.7496 * CONC * SIGX * R2 *
                                            SIGZ/(SIGHCL * E1))
      Y1 = SQRT(AMAX1(Y1,0.0))
C
C         RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C         END OF DFEXP
C
      END
      SUBROUTINE ORGIN(IX0,IY0)
C
C*********************************************************************
C
C         THIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING
C         FOR THE COMPLEX AND MAP SELECTED
C
C*********************************************************************
C
C
C         COMMON BLOCK
C
      COMMON ALT(31),AL1,CONMAX,CONCPK,DEGRAD,ADIR,DOSPK,E1,CLDHT,
             IDIR(31),IOPT(3),ITIME,IDAY,MONTH(2),IYEAR,ISTIM,ISDAY,
             ISMON(2),ISYEAR,IV2,JTOP,JBOT,LAUNTD(10),LTIME,LTIM,LDAY,
             LMON(2),LYEAR,LU,NUM,PI,PIOVR2,PI43,PRESS(31),PTEMP(31),
             SIGHCL,RADDEG,RATOMC,CLDRAD,R2,R3,SAVEA(30),SAVER(30),SIGA,
     .          SIGXO,SIGX,SPEED(31),SQR2PI,SURDEN,SIGZO,SIGAP,SB,TEMP(31),
     .          TOPSUR,TWOPI,ASPD,VPAR(18),CRTIME(31),DIST,YES,Y1,NUMRUN,
     .          YPOS,IFLAG(5),ZB,ZZ,REFLEC,IRETRN
      LOGICAL LTIME
      INTEGER YES
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL203,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION ILINE(32),IDATAF(10),IERS(32),IMAPL(48),NAMEF(3)
C
C         INPUT FORMAT STATEMENT
C
  100 FORMAT (I2,1XA1)
C
C         OUTPUT FORMAT STATEMENT
C
C
C         TYPE AND DIMENSION STATEMENTS
C
      LOGICAL NOT1ST
      DIMENSION IX(8),IY(8)
C
C         DATA STATEMENTS
C
      DATA IERS/32*2H  /
      DATA NAMEF/2H?R,2HIS,2HOP/
      DATA IMAPL/2H40,2H,S,2HEA,2H M,2HAP,2H  ,
     1            2H40,2H,L,2HAN,2HD ,2HMA,2HP ,
     1            2H41,2H,S,2HEA,2H M,2HAP,2H  ,
     1            2H41,2H,L,2HAN,2HD ,2HMA,2HP ,
     1            2H17,2H,S,2HEA,2H M,2HAP,2H  ,
     1            2H17,2H,L,2HAN,2HD ,2HMA,2HP ,
     1            2H39,2H,S,2HEA,2H M,2HAP,2H  ,
     1            2H39,2H,L,2HAN,2HD ,2HMA,2HP /
      DATA NOT1ST/.FALSE./, LCHAR/1HL/
      DATA IX/5450,5411,4830,4825,8750,8730,4100,4100/
      DATA IY/2630,8243,2465,8050,2990,8600,1700,7300/
C
C         IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
C         IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
C         COORDINATES, I, AGAIN
C
      IF(NOT1ST)GO TO 7
C
C         THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
C         AND THE DESIRED MAP, ; *  SEA OR LAND
C
```

```
      NOT1ST = .TRUE.
      CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      IF(YPOS.LT.48.) YPOS = 458.
      IF(IOPT(3).EQ.1) CALL DREAD(NAMEF,8,ILINE)
      IF(IOPT(3).EQ.2) CALL DREAD(NAMEF,9,ILINE)
      IF(IOPT(3).EQ.0) CALL DREAD(NAMEF,10,ILINE)
      CALL LERS(YPOS)
      CALL CHAR(24.,YPOS,0,ILINE(2),8,3,0)
      CALL CHAR(95.,YPOS,0,ILINE(7),50,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IXC,IYC)
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(200.,YPOS+16,0,IERS,25,0,0)
      IF(IXC.LT.6.AND.IOPT(3).EQ.1) I=1
      IF(IXC.GT.5.AND.IXC.LT.12.AND.IOPT(3).EQ.1) I=2
      IF(IXC.GT.11.AND.IXC.LT.18.AND.IOPT(3).EQ.1) I=3
      IF(IXC.GT.17.AND.IOPT(3).EQ.1) I=4
      IF(IXC.LT.6.AND.IOPT(3).EQ.2) I=5
      IF(IXC.GE.6.AND.IOPT(3).EQ.2) I=6
      IF(IXC.LT.6.AND.IOPT(3).EQ.0) I=7
      IF(IXC.GE.6.AND.IOPT(3).EQ.0) I=8
      IMP = (I - 1)*6 + 1
      CALL CHAR(208.,YPOS+16.,0,IMAPL(IMP),12,0,0)
      YPOS = YPOS - 16.
      IF(YPOS.LT.48.) YPOS = 458.
C
C         SET THE COORDINATES BASED ON THE INDEX I
C
    7 IXO = IX(I)
      IYO = IY(I)
C
C         RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C         END OF ORGIN
C
      END
      SUBROUTINE SYMBL(IWIDE,IHI,ISYMB)
      IX=-IWIDE/2
      IY=-IHI/2
      WRITE(12) -1,-1,IX,IY
      WRITE(12,100) IWIDE,0,0,IHI,ISYMB
100   FORMAT(4I5,A1,1H_)
      IY=-IY
      WRITE(12)-1,-1,IX,IY
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
      DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32),IPAR(5)
      CALL RMPAR(IPAR)
      LU = IPAR(1)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
10    CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
100   FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
10    IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS)
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS.LE.48) YPOS = 458.0
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.,YPOS-16.,0,IERS,64,0,0)
      RETURN
      END
      END$
```

Program MIXH

```
FTN4,L
      PROGRAM MIXH
      DIMENSION IPAR(5),Z(20),TV(20),IDCB(256),IBUF(40),FD(20),P(20)
      DIMENSION NAME(3),DUM(20),ITEST(40),T(20),VP(20)
      DIMENSION ITIME(3),IDATE(6)
      DATA C1,C2,C3/-.0005,-.005,100./
      DATA NAME/2H&M,2HIX,2HD1/
      DTV(I) = TV(I+1) - TV(I)
      DZ(I) = Z(I+1) - Z(I)
      GTV(I) = DTV(I) / DZ(I)
      DS(I) = Z(I+1) - Z(1)
      CALL RMPAR(IPAR)
C ** OPEN &MIXD1 DATA FILE
      CALL OPEN(IDCB,IERR,NAME,0)
      LU = IPAR(1)
C   ** INITIALIZE FLAGS TO ZERO
      IFLGI = 0
      IFLBS = 0
      IFLTS = 0
C   ** THIS IS TO INPUT THE TIME AND DATE
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,201) IDATE,ITIME
201   FORMAT(6A2,2X,3A2)
      DO 444 I=1,20
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,*) Z(I),T(I),P(I),FD(I)
C ** CONVERT Z(I) TO METERS
      Z(I) = Z(I)*.3048
C ** CONVERT FD(I) TO DECIMAL
      FD(I) = FD(I)/100.
      VP(I) = 6.11*FD(I)*10.**(7.5*T(I)/(T(I)+237.3))
      TV(I) = (T(I)+273.16)*(1.+.376932*VP(I)/P(I)) -273.16
444   CONTINUE
C ** Z(I) IS ALTITUDE IN METERS
C ** TV(I) IS VIRTUAL TEMPERATURE IN DEG C
C *** P(I) IS PRESSURE IN MILLIBARS
C ** FD(I) IS RELATIVE HUMIDITY
C ** WRITE INPUT VARIABLES
      WRITE(6,6999)
6999  FORMAT(1H1," ALTITUDE  "5X," TEMPERATURE"3X" PRESSURE "5X
     1"RELATIVE HUMIDITY")
      WRITE(6,7000) (Z(I),TV(I),P(I),FD(I),I=1,20)
7000  FORMAT(1H ,4(F10.3,5X))
C ** SPECIFICATION OF HEIGHT OF GROUND BASED INVERSION
      I = 1
      IF(GTV(I).LT.C1) GO TO 2
      DO 11 I = 2,19
      IF(GTV(I).LT.C1) GO TO 12
```

```
11    CONTINUE
      WRITE(6,6000)
6000  FORMAT(//,1H0,"INVALID DATA")
      GO TO 4
12    IF(DS(I).GE.C3) GO TO 100
      GO TO 2
2     WRITE(6,1000)
1000  FORMAT(//,1H0,"NO SURFACE BASED INVERSION")
      IFLGI = 1
      GO TO 4
100   WRITE(6,2000) Z(I)
      GI = Z(I)
      GO TO 4
2000  FORMAT(//,1H ,"TOP OF SURFACE BASED INVERSION = ",F7.2)
C ** SPECIFICATION OF THE BASE OF THE FIRST STABLE LAYER
4     DO 10 I =2,19
      IF(GTV(I).GE.C1) GO TO 60
10    CONTINUE
6     WRITE(6,3000)
      IFLBS = 1
      CALL CLOSE(IDCB,IERR)
      GO TO 9000
3000  FORMAT(//,1H0,"NO STABLE LAYERS")
200   WRITE(6,4000) Z(I)
      BS = Z(I)
      GO TO 30
4000  FORMAT(//,1H0,"BASE OF FIRST STABLE LAYER = ",F7.2)
60    J = I + 1
      DO 61 I = J,19
      IF(GTV(I).GE.C1.AND.DS(I).GE.C3) GO TO 200
61    CONTINUE
      GO TO 6
C ** SPECIFICATION OF THE TOP OF THE FIRST STABLE LAYER
30    J = I + 1
      DO 210 I = J,19
      IF(GTV(I).LE.C2) GO TO 300
210   CONTINUE
      GO TO 400
300   WRITE(6,5000) Z(I)
      TS = Z(I)
5000  FORMAT(//,1H0,"TOP OF FIRST STABLE LAYER = ",F7.2)
      CALL CLOSE(IDCB,IERR)
      GO TO 9000
400   WRITE(6,6001)
6001  FORMAT(//,1H0,"TOP OF STABLE LAYER AT ALTITUDE EXCEEDING THE"
     11X"MAXIMUN ALTITUDE OF AVAILABLE DATA")
      CALL CLOSE(IDCB,IERR)
      IFLTS = 1
C  *** WRITE OUT DATE-TIME FOR GI,BS,TS
9000  CONTINUE
```

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

Section Page

TABLE OF CONTENTS (CONCLUDED)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

```
      WRITE(6,9001)
 9001 FORMAT(1H1," DATE AND TIME "6X"GRND INV"6X"BASE LAYER"6X
     1"TOP LAYER")
      WRITE(6,9002) IDATE,ITIME
 9002 FORMAT(1X,6A2,2X,3A2)
      IF(IFLGI.EQ.1) WRITE(6,9003)
 9003 FORMAT(1H*,22X,"NONE")
      IF(IFLGI.EQ.0) WRITE(6,9004) GI
 9004 FORMAT(1H*,22X,F5.1)
      IF(IFLBS.EQ.1) WRITE(6,9005)
 9005 FORMAT(1H*,36X,"NONE")
      IF(IFLBS.EQ.0) WRITE(6,9006) BS
 9006 FORMAT(1H*,36X,F5.1)
      IF(IFLTS.EQ.1) WRITE(6,9007)
 9007 FORMAT(1H*,53X,"NONE")
      IF(IFLTS.EQ.0) WRITE(6,9008) TS
 9008 FORMAT(1H*,53X,F7.2)
      END
      END$
```

Program JWSPL

```
FTN4
      PROGRAM JWSPL
      COMMON IDATE(8),ITIME(3),IDCB(144),IBUF(40),LU,IDSAV(8),ISAVF
     1,NEWD
      DIMENSION IAX1(11), XY1(3), XY2(3),IYTIT(21),ISDAT(8)
      DIMENSION WS1(800),WS2(800)
      DIMENSION IYLAB(8),IXLAB(12),IXLAB3(2),NTIME(4)
      DIMENSION IXLAB2(3),OKPLT(3),IXDATE(9),IXTIT(21)
      DIMENSION ILEG1(7),ILEG2(14),ILEG3(14)
      DIMENSION NUMFB(2),IPAR(5), NAME(3),NAME1(3), NAME2(3), NAME3(3)
      DATA ILEG1/12,2HNA,2HSA,2H -,2H M,2HSF,2HC /
      DATA  XY1/0.,-.3,-.6/
      DATA  XY2/0.,.3,.6/
      DATA ILEG2/26,2HSP,2HAC,2HE ,2HSC,2HIE,2HNC,2HES,
     12H L,2HAB,2HOR,2HAT,2HOR,2HY /
      DATA IAX1/2,2H 2,2H 4,2H 6,2H 8,2H10,2H12,2H14,2H16,2H18,2H20/
      DATA ILEG3/26,2HAE,2HRO,2HSP,2HAC,2HE ,2HEN,2HVI,2HRO,
     12HNM,2HEN,2HT ,2HDI,2HV /
      DATA IYLAB/14,2HAL,2HTI,2HTU,2HDE,2H (,2HKM,2H) /
      DATA NTIME/5/
      DATA IXLAB/21,2HSC,2HAL,2HAR,2H W,2HIN,2HD ,2HSP,2HEE,2HD ,
     1       2H(M,2HS /
      DATA IXLAB2/2,2H-1/
      DATA IXLAB3/1,2H) /
      DATA IXTIT/40,2H  ,2HPO,2HIN,2HT ,2HMU,2HGU,2H J,2HIM,2HSP,
     1      2HHE,2HRE,2H W,2HIN,2HD ,2HPR,2HOF,2HIL,2HE ,2HDA,2HTA/
      DATA IYTIT/40,2HCA,2HPE,2H K,2HEN,2HNE,2HDY,2H J,2HIM,2HSP,
     1      2HHE,2HRE,2H W,2HIN,2HD ,2HPR,2HOF,2HIL,2HE ,2HDA,2HTA/
      DATA IXDATE/16/
      DATA NAME1/2H&J,2HKS,2HC1/
      DATA NAME2/2H&J,2HKS,2HC2/
      DATA NAME3/2H&J,2HPT,2HM /
C ** INITIALIZE LU  DEVICE
      LU = 7
C ** OPEN DATE AND TIME FILE
C     CALL OPEN(IDCB,IERR,NAME,0)
C ** INITIALIZE PLOTTER
      CALL PLTLU(12)
      NEWD = 0
      IF1 = 0
      INAME = 0
      ISAVF = 0
      IPDY = 0
      CALL CLEER
      WRITE(LU,405)
405   FORMAT(//"****NASA/MSFC JIMSPHERE WIND PROFILE PLOTTING"
     11X"PROGRAM****")
      WRITE(LU,214)
214   FORMAT(/"Jimsphere Wind Profile Data Desired?"5X"WIND SPEED"
     110X"WIND DIRECTION")
```

```
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.10) GO TO 215
      WRITE(LU,216)
216   FORMAT(/"Use Program JWDPL For Wind Directio. Plots")
      WRITE(LU,217)
217   FORMAT(/," **** JWSPL *** TERMINATED ****")
      STOP
215   CONTINUE
      WRITE(LU,102)
102   FORMAT(/,"Jimsphere Wind Profile Data Desired?"5X"CAPE KENNEDY"
     110X"POINT MUGU")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.GT.10) GO TO 71
      ICK = 1
      WRITE(LU,103)
103   FORMAT(/,5X,"Cape Kennedy Data Desired?"10X"1964-1966"
     19X"1967-1970")
104   FORMAT(/,10X"Point Mugu Data For:  1965-1970")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LE.9) WRITE(LU,105)
      IF(IX.LE.9) INAME = 1
      IF(IX.GT.9) WRITE(LU,106)
      IF(IX.GT.9) INAME = 2
      IF(INAME.EQ.2) GO TO 172
      DO 141 K=1,3
      NAME(K) = NAME1(K)
141   CONTINUE
      GO TO 173
172   CONTINUE
      DO 142 K = 1,3
      NAME(K) = NAME2(K)
142   CONTINUE
173   CONTINUE
105   FORMAT(/,10X"Cape Kennedy Data For:  1964-1966")
106   FORMAT(/,10X"Cape Kennedy Data For:  1967-1970")
      GO TO 72
71    CONTINUE
      DO 171 J=1,3
      NAME(J) = NAME3(J)
171   CONTINUE
      WRITE(LU,104)
72    CONTINUE
C     WRITE(LU,108)
C108  FORMAT(//,"Jimsphere Wind Speed Data Being Processed")
```

```
      WRITE(LU,400)
400   FORMAT(//,"TURN ON PLOTTER....POSITION PAPER....TOUCH PANEL WHEN"
     11X"READY")
234   CONTINUE
      XX1 = 56.
      XX1 = 34.
      XX1 = 30.
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.GT.15) GO TO 234
C THIS IS WHERE THE DISC FILE IS OPENED
      CALL OPEN(IDCB,IERR,NAME,0)
      CALL CLEER
      WRITE(LU,907)
907   FORMAT(/"****PLOTTING HAS BEEN INITIALIZED****")
941   CONTINUE
69    IFLAG = 0
      CALL LLEFT
      CALL SFACT(15.,10.)
      CALL PLOT(1.,1.5,-3)
C ** WRITE NASA LEGEND
      CALL PLOT(0.,0.,3)
      CALL PLOT(-.5,-.95,3)
      CALL SYMB(-.5,-.95,.1,ILEG1,0.,1)
      CALL PLOT(-.5,-1.1,3)
      CALL SYMB(-.5,-1.1,.08,ILEG2,0.,1)
      CALL PLOT(-.5,-1.25,3)
      CALL SYMB(-.5,-1.25,.08,ILEG3,0.,1)
C ** THIS PORTION DRAWS Y-AXIS
      CALL PLOT(0.,0.,3)
      CALL PLOT(0.,0.,2)
      CALL PLOT(0.,5.,2)
      DO 30 I=1,10
      A = I/2.
      CALL PLOT(0.,A,3)
      CALL PLOT(.05,A,2)
      B = I*2
      CALL NUMB(-.3,A,.1,B,0.,-1)
30    CONTINUE
      CALL SYMB(-.45,1.9,.10,IYLAB,90.,1)
C ** THIS PORTION WRITES HEADERS AND LEGEND
      CALL SYMB(3.5, -1.1,.10,IXLAB,0.,1)
      CALL SYMB(5.6,-1.0,.1,IXLAB2,0.,1)
      CALL SYMB(5.8,-1.1,.10,IXLAB3,0.,1)
      IF(INAME.EQ.0) CALL SYMB(2.3,6.0,.12,IXTIT,0.,1)
      IF(INAME.GT.0) CALL SYMB(2.3,6.0,.12,IYTIT,0.,1)
C ** THIS PORTION READS THE FIRST WS1 DATA ARRAY
      IF(IF1.EQ.0) CALL RWS2(WS1,IFLAG)
      IF(NEWD.EQ.1) GO TO 941
      IF1 = 1
      IF(ISAVF.NE.1) GO TO 129
      DO 128 J=1,8
      IDATE(J) = IDSAV(J)
128   CONTINUE
      IF(ISAVF.EQ.1) ISAVF = 0
129   CONTINUE
      IFLAG = 0
      DO 571 KL=1,8
      KP = KL + 1
      IXDATE(KP) = IDATE(KL)
571   CONTINUE
      CALL SYMB(4.,5.70,.12,IXDATE,0.,1)
      X = 0.
      XY11 = XY1(1)
      XY22 = XY2(1)
      XYFLG = 1
C ** THIS PORTION DRAWS THE X AXIS *********
95    CALL PLOT(0.,0.,3)
      CALL PLOT(X,XY11,-3)
      XX=0.
      DO 456 I=1,799
      IF(WS1(I).GE.100.) GO TO 456
      XX = AMAX1(XX,WS1(I))
456   CONTINUE
      IX = XX/10 + 2
      IF(IX.GT.6) IX = 6
      XI = .5 + (IX - 2)* .5
      CALL PLOT(XI,0.,2)
      DO 35 I=1,IX
      A = (I-1)/2.
      D = (I-1)*10.
      CALL PLOT(A,0.,3)
      CALL PLOT(A,.05,2)
      B = A - .05
      CALL NUMB(B,-.15,.1,D,0.,-1)
35    CONTINUE
      B=0.
      JC = 0
      CALL PLOT(0.,XY22,-3)
      IF(WS1(1).GE.100.) GO TO 642
      A=WS1(1)/20.
      B= B + .00625
      CALL PLOT(A,B,3)
642   CONTINUE
      DO 36 I=2,799
      B = B + .00625
      IF(WS1(I).GE.100.) GO TO 643
      A = WS1(I)/20.
      JC = JC + 1
      IF(JC.EQ.1) CALL PLOT(A,B,3)
      CALL PLOT(A,B,2)
      CC= B
643   CONTINUE
36    CONTINUE
      C = A -.25
      D = CC+ .05
      NTIME(2) = ITIME(1)
      NTIME(3) = ITIME(2)
      NTIME(4) = ITIME(3)
      CALL SYMB(C,D,.08,NTIME,0.,1)
      CALL RWS2(WS2,IFLAG)
      IF(NEWD.EQ.1) GO TO 941
      IF(IFLAG.EQ.0) GO TO 70
      DO 96 I=1,799
      WS1(I) = WS2(I)
96    CONTINUE
      GO TO 69
70    X = 0.
      DO 300 I=1,799
      IF(WS1(I).GE.100.OR.WS2(I).GE.100.) DIFF    = 0.
      IF(WS1(I).GE.100.OR.WS2(I).GE.100.) GO TO 300
      DIFF    = WS1(I) - WS2(I)
      X = AMAX1(X,DIFF)
300   CONTINUE
      X = (X/20.)
      IF(X.LE..5) X = 0.5
      IF(X.GT..5.AND.X.LE.1.) X = 1.0
      IF(X.GT.1..AND.X.LE.1.5) X = 1.5
      IF(X.GT.1.5.AND.X.LE.2.) X = 2.0
      IF(X.GT.2..AND.X.LE.2.5) X = 2.5
      X = X + 0.5
      IF(XYFLG.EQ.1) XY11 = XY1(2)
      IF(XYFLG.EQ.1) XY22 = XY2(2)
      IF(XYFLG.EQ.2) XY22 = XY2(3)
      IF(XYFLG.EQ.2) XY11 = XY1(3)
      IF(XYFLG.EQ.3) XY11 = XY1(1)
      IF(XYFLG.EQ.3) XY22 = XY2(1)
      IF(XYFLG.EQ.3) XYFLG = 0
      IF(XYFLG.EQ.3) CALL PLOT(0.,.6,-3)
      XYFLG = XYFLG + 1
      DO 80 I=1,799
      WS1(I) = WS2(I)
80    CONTINUE
      GO TO 95

999   CALL WRITE
      CALL CLOSE(IDCB,IERR)
      STOP
      END
      SUBROUTINE CLEER
      INTEGER RSFF
      DATA RSFF/017014B/
      CALL EXEC(2,107B,RSFF,-2)
      RETURN
      END
      SUBROUTINE RWS2(WS2,IFLAG)
      COMMON IDATE(8),ITIME(3),IDCB(144),IBUF(40),LU,IDSAV(8),ISAVF
     1,NEWD
      DIMENSION WS2(800)
      DATA IBLK/2H  /
      ICF = 0
      IF(NEWD.EQ.1) GO TO 942
15    IK = 1
      DO 51 K = 1,100
      KK = K*8
      READ(8,*) (WS2(IJ),IJ=IK,KK)
      IK = IK + 8
51    CONTINUE
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,300)(IDATE(NN),NN=1,8),(ITIME(NK),NK=1,3)
300   FORMAT(8A2,3X,3A2)
      IF(IDATE(1).EQ.IBLK) GO TO 20
      DO 89 J=1,8
      IDSAV(J) = IDATE(J)
89    CONTINUE
      IF(ICF.EQ.1) GO TO 953
      IF(IRDY.EQ.0) GO TO 45
      CALL CLEER
      WRITE(LU,580)
580   FORMAT(/,"DO YOU WISH TO TERMINATE PROGRAM?"10X"YES"10X"NO")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.10) WRITE(LU,349)
      IF(IX.LT.10) STOP
349   FORMAT(/"****PROGRAM JIMPL HAS BEEN TERMINATED****")
      WRITE(LU,101)
101   FORMAT(//"CHANGE PLOT PAPER.............TOUCH PANEL TO CONTINUE")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.15) IFLAG = 1
45    CONTINUE
      IRDY = 1
      WRITE(LU,100) (IDATE(NK),NK=1,8)
100   FORMAT(//"New Date is:      "8A2)
C     RETURN
      WRITE(LU,940)
```

```
940   FORMAT(/,"Different Date Desired?"10X"YES"10X"NO")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.GT.6) GO TO 942
      WRITE(LU,951)
951   FORMAT(/"ENTER NUMBER OF CURVES SKIPPED? _")
      READ(LU,*) ICURS
      NFB = ICURS - 1
      IFB = NFB
      NFB = NFB*100
      CALL PTAPE(8,0,NFB)
      CALL POSNT(IDCB,IERR,IFB,0)
      ICF = 1
      GO TO 15
953   CONTINUE
942   CONTINUE
      NEWD = 0
      ISAVF = 1
20    CONTINUE
      WRITE(LU,301)(ITIME(NK),NK=1,3)
301   FORMAT(//"Time of Curve is: "3A2,5X"Plot Desired?"10X"YES"10X"NO")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.GT.10) WRITE(LU,223)
223   FORMAT(10X"CURVE NOT PLOTTED")
      IF(IX.GT.10)  GO TO 15
      IF(IX.LE.10) WRITE(LU,222)
222   FORMAT(/,"Curve Desired....Will It Fit On Paper?"5X"YES"10X"NO")
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.10) GO TO 23
      CALL CLEER
      WRITE(LU,101)
      CALL TOUCH(0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.15) ISAVF = 1
      IF(IX.LT.15) IFLAG = 1
23    CONTINUE
      WRITE(LU,414)
414   FORMAT(5X,"Curve Being Plotted")
      CALL FILTR(WS2)
      RETURN
      END
      SUBROUTINE TOUCH(IXL,IXH,IYL,IYH,IX,IY)
      INTEGER ENQ
      DIMENSION I(2)
      EQUIVALENCE (IA,I(1)),(IB,I(2))
      DATA ENQ/002400B/
    4 CALL EXEC(2,107B,ENQ,-1)
      CALL EXEC(1,107B,I,-4)
      IX = IAND(I(1),37B)
      IY = IAND(ISHIF(I(2),8),37B)
      RETURN
      END
      SUBROUTINE FILTR(WS1)
      DIMENSION WS1(1)
      DO 1000 IC1=1,798
      IC2 = IC1 + 1
      IC3 = IC1 + 2
      IF(WS1(IC1) .GE. 100.0)GO TO 1000
      DIF1 = WS1(IC1) - WS1(IC2)
      DIF2 = WS1(IC1) - WS1(IC3)
      DIF3 = WS1(IC2) - WS1(IC3)
      IF(ABS(DIF1).GT.1.0 .AND. ABS(DIF3).GT.1.0)WS1(IC2) = WS1(IC1)
      IF((ABS(DIF1).GT.1.0) .AND. (ABS(DIF2).GT.1.0) .AND.
     .   (((DIF1.GT.0.0) .AND. (DIF3.LT.0.0)) .OR.
     .     ((DIF1.LT.0.0) .AND. (DIF3.GT.0.0))))WS1(IC2) = WS1(IC1)
 1000 CONTINUE
      RETURN
      END
      END$
```

Program JWDPL

```
FTN4,L
      PROGRAM JUDPL
      COMMON IDATE(8),ITIME(3),IDCB(144),IBUF(40),LU,IDSAV(8),ISAVF
     1,NEUD
      DIMENSION IAX1(11), XY1(3), XY2(3),IYTIT(21),ISDAT(8)
      DIMENSION US1(800),US2(800),DEG1(12),DEG2(12),DEG3(12),DEG4(12)
      DIMENSION IYLAB(8),IXLAB(12),IXLAB3(2),NTIME(4)
      DIMENSION IXLAB2(3),OKPLT(3),IXDATE(9),IXTIT(21)
      DIMENSION SKP(8)
      DIMENSION ILEG1(7),ILEG2(14),ILEG3(14)
      DIMENSION NUMFB(2),IPAR(5), NAME(3),NAME1(3), NAME2(3), NAME3(3)
      DATA DEG1/0.,90.,180.,270.,360.,90.,180.,270.,360.,90.,180.,270./
      DATA SKP/270.,180.,90.,0.,-90.,-180.,-270.,360./
      DATA DEG2/90.,180.,270.,360.,90.,180.,270.,360.,90.,180.,270.,
     1360./
      DATA DEG3/180.,270.,360.,90.,180.,270.,360.,90.,180.,270.,360.,
     190./
      DATA DEG4/270.,360.,90.,180.,270.,360.,90.,180.,270.,360.,90.,
     1180./
      DATA ILEG1/12,2HNA 2HSA,2H -,2H M,2HSF,2HC /
      DATA  XY1/0.,-.3,-.6/
      DATA  XY2/0.,.3,.6/
      DATA ILEG2/26,2HSP,2HAC,2HE ,2HSC,2HIE,2HNC,2HES,
     12H L,2HAB,2HOR,2HAT,2HOR,2HY /
      DATA IAX1/2,2H 2,2H 4,2H 6,2H 8,2H10,2H12,2H14,2H16,2H18,2H20/
      DATA ILEG3/26,2HAE,2HRO,2HSP,2HAC,2HE ,2HEN,2HVI,2HRO,
     12HNM,2HEN,2HT ,2HDI,2HV./
      DATA IYLAB/14,2HAL,2HTI,2HTU,2HDE,2H (,2HKM,2H) /
      DATA NTIME/5/
      DATA IXLAB/21,2HWI,2HND,2H D,2HIR,2HEC,2HTI,2HON,2H (,2HDE,
     1       2HGS,2H) /
      DATA IXLAB2/2,2H-1/
      DATA IXLAB3/1,2H) /
      DATA IXTIT/40,2H  ,2HPO,2HIN,2HT ,2HMU,2HGU,2H J,2HIM,2HSP,
     1      2HME,2HRE,2H W,2HIN,2HD ,2HPR,2HOF,2HIL,2HE ,2HDA,2HTA/
      DATA IYTIT/40,2HCA,2HPE,2H K,2HEN,2HNE,2HDY,2H J,2HIM,2HSP,
     1      2HME,2HRE,2H W,2HIN,2HD ,2HPR,2HOF,2HIL,2HE ,2HDA,2HTA/
      DATA IXDATE/16/
      DATA NAME1/2H&J,2HKS,2HC1/
      DATA NAME2/2H&J,2HKS,2HC2/
      DATA NAME3/2H&J,2HPT,2HM /
C ** INITIALIZE LU  DEVICE
      CALL RMPAR(IPAR)
      LU = IPAR(1)
C     CALL EXEC(22,1)
      LU = 7
C ** OPEN DATE AND TIME FILE
C     CALL OPEN(IDCB,IERR,NAME,0)
C ** INITIALIZE PLOTTER
      CALL PLTLU(12)
      NEWD = 0
      IF1 = 0
      INAME = 0
      ISAVF = 0
      IRDY = 0
      CALL CLEAR
C     WRITE(LU,405)
C405  FORMAT(//"****NASA/MSFC JIMSPHERE WIND PROFILE PLOTTING"
C    11X"PROGRAM****")
C     WRITE(LU,214)
C214  FORMAT(/"Jimsphere Wind Profile Data Desired?"5X"WIND SPEED"
C    110X"WIND DIRECTION")
C     CALL TOUCH(0,15,0,15,IX,IY)
C     IF(IX.GT.9)  GO TO 215
C     WRITE(LU,216)
C16   FORMAT(/"Use Program JWDPL For Wind Direction Plots")
C     WRITE(LU,217)
C17   FORMAT(/," **** JWSPL *** TERMINATED ****")
C     STOP
215   CONTINUE
      WRITE(LU,102)
102   FORMAT(/,"Jimsphere Wind Profile Data Desired?"5X"CAPE KENNEDY"
     110X"POINT MUGU")
      CALL TOUCH(0,15,0,15,IX,IY)
      IF(IX.GT.10) GO TO 71
      ICK = 1
      WRITE(LU,103)
103   FORMAT(/,5X,"Cape Kennedy Data Desired?"10X"1964-1966"
     19X"1967-1970")
104   FORMAT(/,10X"Point Mugu Data For:  1965-1970")
      CALL TOUCH(0,15,0,15,IX,IY)
      IF(IX.LE.9) WRITE(LU,105)
      IF(IX.LE.9) INAME = 1
      IF(IX.GT.9) WRITE(LU,106)
      IF(IX.GT.9) INAME = 2
      IF(INAME.EQ.2) GO TO 172
      DO 141 K=1,3
      NAME(K) = NAME1(K)
141   CONTINUE
      GO TO 173
172   CONTINUE
      DO 142 K = 1,3
      NAME(K) = NAME2(K)
142   CONTINUE
173   CONTINUE
105   FORMAT(/,10X"Cape Kennedy Data For:  1964-1966")
106   FORMAT(/,10X"Cape Kennedy Data For:  1967-1970")
      GO TO 72
71    CONTINUE
      DO 171 J=1,3
      NAME(J) = NAME3(J)
171   CONTINUE
      WRITE(LU,104)
72    CONTINUE
C     WRITE(LU,108)
C108  FORMAT(//,"Jinsphere Wind Speed Data Being Processed")
      WRITE(LU,400)
400   FORMAT(//,"TURN ON PLOTTER....POSITION PAPER....TOUCH PANEL WHEN"
     11X"READY")
234   CALL TOUCH(0,15,0,15,IX,IY)
      IF(IX.GT.15) GO TO 234
C THIS IS WHERE THE DISC FILE IS OPENED
      CALL OPEN(IDCB,IERR,NAME,0)
      CALL CLEAR
C     WRITE(LU,907)
C07   FORMAT(/"****PLOTTING HAS BEEN INITIALIZED****")
941   CONTINUE
69    IFLAG = 0
      CALL LLEFT
      CALL SFACT(15.,10.)
      CALL PLOT(1.,1.5,-3)
C ** WRITE NASA LEGEND
      CALL PLOT(0.,0.,3)
      CALL PLOT(-.5,-.95,3)
      CALL SYMB(-.5,-.95,.1,ILEG1,0.,1)
      CALL PLOT(-.5,-1.1,3)
      CALL SYMB(-.5,-1.1,.08,ILEG2,0.,1)
      CALL PLOT(-.5,-1.25,3)
      CALL SYMB(-.5,-1.25,.08,ILEG3,0.,1)
C ** THIS PORTION DRAWS Y-AXIS
      CALL PLOT(0.,0.,3)
      CALL PLOT(0.,0.,2)
      CALL PLOT(0.,5.,2)
      DO 30 I=1,10
      A = I/2.
      CALL PLOT(0.,A,3)
      CALL PLOT(.05,A,2)
      B = I*2
      CALL NUMB(-.3,A, 1,B,0.,-1)
30    CONTINUE
      CALL SYMB(-.45,1.9,.10,IYLAB,90.,1)
C ** THIS PORTION WRITES HEADERS AND LEGEND
      CALL SYMB(3.5, -1.1,.10,IXLAB,0.,1)
      IF(INAME.EQ.0) CALL SYMB(2.3,6.0,.12,IXTIT,0.,1)
      IF(INAME.GT.0) CALL SYMB(2.3,6.0,.12,IYTIT,0.,1)
C ** THIS PORTION READS THE FIRST WS1 DATA ARRAY
      IF(IF1.EQ.0) CALL RWS2(WS1,IFLAG,IQDS)
      IF(NEWD.EQ.1) GO TO 941
      IF1 = 1
      IF(ISAVF.NE.1) GO TO 129
      DO 128 J=1,8
      IDATE(J) = IDSAV(J)
128   CONTINUE
      IF(ISAVF.EQ.1) ISAVF = 0
129   CONTINUE
      IFLAG = 0
      DO 571 KL=1,8
      KP = KL + 1
      IXDATE(KP) = IDATE(KL)
571   CONTINUE
      CALL SYMB(4.,5.70,.12,IXDATE,0.,1)
      X = 0.
      XY11 = XY1(1)
      XY22 = XY2(1)
      XYFLG = 1
C ** THIS PORTION DRAWS THE X AXIS *********
95    CALL PLOT(0.,0.,3)
      CALL PLOT(X,XY11,-3)
      XX=0.
      DO 456 I=1,790
      IF(WS1(I).GE.1000.) GO TO 456
      XX = AMAX1(XX,WS1(I))
456   CONTINUE
      XX=(XX+SKP(IQDS))/180.0
      XX=AINT(2.0*(XX+0.49999999))/2.0
      CALL PLOT(XX,0.,2)
      NXX = INT(2.0 * XX) + 1
      DO 35 I=1,NXX
      A = (I-1)/2.
      D=DEG1(I)
      IF(IQDS.EQ.1) D = DEG2(I)
      IF(IQDS.EQ.2) D = DEG3(I)
      IF(IQDS.EQ.3) D = DEG4(I)
      IF(IQDS.EQ.4) D = DEG1(I)
      IF(IQDS.EQ.5) D = DEG2(I)
      IF(IQDS.EQ.6) D = DEG3(I)
      IF(IQDS.EQ.7) D = DEG4(I)
      IF(IQDS.EQ.8) D = DEG1(I)
      CALL PLOT(A,0.,3)
      CALL PLOT(A,.05,2)
      B = A - .05
      CALL NUMB(B,-.15,.1,D,0.,-1)
35    CONTINUE
      B=0.
      CALL PLOT(0.,XY22,-3)
      IF(WS1(1).GE.1000.) GO TO 642
      A=(WS1(1)+SKP(IQDS))/180.
      B= B + .00625
      CALL PLOT(A,B,3)
642   CONTINUE
```

```
      N = 1
      JFLG = -1
      DO 36 I=2,790
      B = B + .00625
      IF(WS1(I).GE.1000.) GO TO 643
      A = (WS1(I)+SKP(IQDS))/180
      XC = ABS(WS1(I)-WS1(N))
      IF(JFLG.NE.0) CALL PLOT(A,B,3)
      JFLG=0
      CALL PLOT(A,B,2)
      CC= B
      N = I
643   CONTINUE
36    CONTINUE
      C = A -.25
      D = CC+ .05
      NTIME(2) = ITIME(1)
      NTIME(3) = ITIME(2)
      NTIME(4) = ITIME(3)
      CALL SYMB(C,D,.08,NTIME,0.,1)
      CALL RWS2(WS2,IFLAG,IQDS)
      IF(NEWD.EQ.1) GO TO 941
      IF(IFLAG.EQ.0) GO TO 70
      DO 96 I=1,790
      WS1(I) = WS2(I)
96    CONTINUE
      GO TO 69
70    X = XX
      IF(XYFLG.EQ.1) XY11 = XY1(2)
      IF(XYFLG.EQ.1) XY22 = XY2(2)
      IF(XYFLG.EQ.2) XY22 = XY2(3)
      IF(XYFLG.EQ.2) XY11 = XY1(3)
      IF(XYFLG.EQ.3) XY11 = XY1(1)
      IF(XYFLG.EQ.3) XY22 = XY2(1)
      IF(XYFLG.EQ.3) XYFLG = 0
      IF(XYFLG.EQ.3) CALL PLOT(0.,.6,-3)
      XYFLG = XYFLG + 1
      DO 80 I=1,790
      WS1(I) = WS2(I)
80    CONTINUE
      GO TO 95

999   CALL WRITE
C     CALL EXEC(22,0)
      CALL CLOSE(IDCB,IERR)
      END
      SUBROUTINE RWS2(WS2,IFLAG,IQDS,DSHF)
      COMMON IDATE(8),ITIME(3),IDCB(144),IBUF(40),LU,IDSAV(8),ISAVF
     1,NEWD
      DIMENSION WS2(800)
      DATA IBLK/2H  /
      ICF = 0
      IF(NEWD.EQ.1) GO TO 942
15    IK = 1
      DO 51 K = 1,100
      KK = K*8
      READ(8,*) (WS2(IJ),IJ=IK,KK)
      IK = IK + 8
51    CONTINUE
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,300)(IDATE(NN),NN=1,8),(ITIME(NK),NK=1,3)
300   FORMAT(8A2,3X,3A2)
      IF(IDATE(1).EQ.IBLK) GO TO 20
      DO 89 J=1,8
      IDSAV(J) = IDATE(J)
89    CONTINUE
      IF(ICF.EQ.1) GO TO 953
      IF(IRDY.EQ.0) GO TO 45
      CA\L CLEAR
      WRITE(LU,580)
580   FORMAT(/,"DO YOU WISH TO TERMINATE PROGRAM?"10X"YES"10X"NO")
      CALL TOUCH(0,15,0,15,IX,IY)
C     IF(IX.LT.10) WRITE(LU,349)
      IF(IX.LT.10) STOP
C49   FORMAT(/"****PROGRAM JIMPL HAS BEEN TERMINATED****")
      WRITE(LU,101)
101   FORMAT(//"CHANGE PLOT PAPER............TOUCH PANEL TO CONTINUE")
      CALL TOUCH(0,15,0,15,IX,IY)
      IF(IX.LT.15) IFLAG = 1
45    CONTINUE
      IRDY = 1
      WRITE(LU,100) (IDATE(NK),NK=1,8)
100   FORMAT(//"New Date is:      "8A2)
C     RETURN
      WRITE(LU,940)
940   FORMAT(/,"Different Date Desired?"10X"YES"10X"NO")
      CALL TOUCH(0,15,0,15,IX,IY)
      IF(IX.GT.6) GO TO 942
      WRITE(LU,951)
951   FORMAT(/"ENTER NUMBER OF CURVES SKIPPED? _")
      READ(LU,*) ICURS
      NFB = ICURS - 1
      IFB = NFB
      NFB = NFB*100
      CALL PTAPE(8,0,NFB)
      CALL POSNT(IDCB,IERR,IFB,0)
      ICF = 1
      GO TO 15
953   CONTINUE
```

```
942     CONTINUE
        NEWD = 0
        ISAVF = 1
20      CONTINUE
        WRITE(LU,301)(ITIME(NK),NK=1,3)
301     FORMAT(//"Time of Curve is: "2A2,A1,5X"Plot Desired?"10X"YES"
                10X"NO")
        CALL TOUCH(0,15,0,15,IX,IY)
C       IF(IX.GT.10) WRITE(LU,223)
C23     FORMAT(10X"CURVE NOT PLOTTED")
        IF(IX.GT.10)  GO TO 15
        IF(IX.LE.10) WRITE(LU,222)
222     FORMAT(/,"Curve Desired....Will It Fit On Paper?"5X"YES"10X"NO")
        CALL TOUCH(0,15,0,15,IX,IY)
        IF(IX.LT.10) GO TO 23
        CALL CLEAR
        WRITE(LU,101)
        CALL TOUCH(0,15,0,15,IX,IY)
        IF(IX.LT.15) ISAVF = 1
        IF(IX.LT.15) IFLAG = 1
23      CONTINUE
C       WRITE(LU,414)
C14     FORMAT(5X,"Curve Being Plotted")
        CALL FILTR(WS2,IQDS,DSHF)
        RETURN
        END
        SUBROUTINE TOUCH(IXL,IXH,IYL,IYH,IX,IY)
        INTEGER ENQ
        DIMENSION I(2)
        EQUIVALENCE (IA,I(1)),(IB,I(2))
        DATA ENQ/002400B/
      4 CALL EXEC(2,107B,ENQ,-1)
        CALL EXEC(1,107B,I,-4)
        IX = IAND(ISHIF(IB,-8),1B)
        IX = IOR(IAND(ISHIF(IA,-3),2B),IX)
        IX = IOR(IAND(ISHIF(IA,-1),4B),IX)
        IX = IOR(IAND(ISHIF(IA,1),10B),IX)
        IF(IX.LT.IXL .OR. IX.GT.IXH)GO TO 4
        IY = IAND(ISHIF(IB,-12),1B)
        IY = IOR(IAND(ISHIF(IB,-10),2B),IY)
        IY = IOR(IAND(ISHIF(IB,-8),4B),IY)
        IY = IOR(IAND(ISHIF(IB,-6),10B),IY)
        IF(IY.LT.IYL .OR. IY.GT.IYH)GO TO 4
        RETURN
        END
        SUBROUTINE CLEAR
        INTEGER RSFF
        DATA RSFF/017014B/
        CALL EXEC(2,107B,RSFF,-2)
        RETURN
      END
      SUBROUTINE FILTR(WS1,IQDS,DSHF)
      DIMENSION WS1(1)
      N = 1
      DO 100 I=2,790
      IF(WS1(N).GT.1000.) GO TO 50
      IF(WS1(I).GT.1000.) GO TO 100
      IF(ABS(WS1(N)-WS1(I)).LT.180.)  GO TO 50
      IF(WS1(I).GT.WS1(N))GO TO 40
      WS1(I) = WS1(I) + 360.0
      GO TO 50
   40 WS1(I) = WS1(I) - 360.0
50    N = I
100   CONTINUE
      RMIN = 10000.
      DO 101 I=2,790
      RMIN = AMIN1(RMIN,WS1(I))
101   CONTINUE
      IF(RMIN.GT.-270..AND.RMIN.LE.-180.) IQDS = 1
      IF(RMIN.GT.-180..AND.RMIN.LE.-90. ) IQDS = 2
      IF(RMIN.GT.-90. .AND.RMIN.LE.0.   ) IQDS = 3
      IF(RMIN.GT.0.   .AND.RMIN.LE.90.  ) IQDS = 4
      IF(RMIN.GT.90.  .AND.RMIN.LE  180.) IQDS = 5
      IF(RMIN.GT.180. .AND.RMIN.LE.270. ) IQDS = 6
      IF(RMIN.GT.270. .AND.RMIN.LE.1000. ) IQDS = 7
      IF(RMIN.GT.-360. .AND. RMIN.LE.-270.) IQDS = 8
      IF(IQDS.EQ.1) DSHF = 270.
      IF(IQDS.EQ.2) DSHF = 180.
      IF(IQDS.EQ.3) DSHF = 90.
      IF(IQDS.EQ.4) DSHF = 0.
      IF(IQDS.EQ.5) DSHF = -90.
      IF(IQDS.EQ.6) DSHF = -180.
      IF(IQDS.EQ.7) DSHF = -270.
      IF(IQDS.EQ.8) DSHF = 360.
      RETURN
      END
      SUBROUTINE IQDCP(IQD,A)
      IF(IQD.EQ.2.AND.A.GT..5) A = A - .5
      IF(IQD.EQ.2.AND.A      .LE..5) A = A + 1.5
      IF(IQD .EQ.3.AND.A.LE.1.)A=A+1.
      IF(IQD .EQ.3.AND.A.GT.1.) A = A-1.
      IF(IQD .EQ.4.AND.A.LE.1.5)A=A+.5
      IF(IQD .EQ.4.AND.A.GT.1.5)A=A-1.5
      RETURN
      END
      END$
```

Program JIMPS

```
FTN4
      PROGRAM JIMPS
      DIMENSION X(4),Y(4),XX(10),YY(800),IHEAD(20),WS1(800)
      DIMENSION XTIC(12),XTIC1(2), YTIC(42),YTIC1(2),IXNUM(6)
      DIMENSION IPAR(5)
      DIMENSION ITIME(18),IERS(32),JWSPL(3),ILINE(32)
      DIMENSION IXLEG(9),JIMF1(3)
      DIMENSION IDATE(8),YLAB(10),YALT(8),XNUM(12)
      INTEGER YLAB,YALT
      REAL LL
      DATA IERS/32*20040B/
      DATA JWSPL/2HJW,2HSP,2HL /
      DATA JIMF1/2HJI,2HMF,2H1 /
      DATA ITIME/2H13,2H06,2Hz ,2H14,2H31,2Hz ,2H16,2H00,2Hz ,
     12H17,2H31,2Hz ,2H19,2H00,2Hz ,2H22,2H00,2Hz /
      DATA X/64.,164.,64.,64./
      DATA YTIC1/62.,66./
      DATA YTIC/152.,152.,176.,176.,200.,200.,224.,224.,248.,248.,
     1272.,272.,296.,296.,320.,320.,344.,344.,368.,368./
      DATA IXNUM/2H 0,2H10,2H20,2H30,2H40,2H50/
      DATA IXLEG/2HSc,2Hal,2Har,2H W,2Hin,2Hd ,2HSp,2Hee,2Hd /
      DATA XNUM/56.,56.,76.,76.,96.,96.,116.,116.,136.,136.,
     1156.,156./
      DATA  XTIC/64.,64.,84.,84.,104.,104.,124.,124.,144.,144.,
     1164.,164./
      DATA XTIC1/126.,130./
      DATA IHEAD/2HCa,2Hpe,2H K,2Hen,2Hne,2Hdy,2H J,2Him,2Hsp,2Hhe,
     12Hre,2H W,2Hin,2Hd ,2HPr,2Hof,2Hil,2He ,2HDa,2Hta/
      DATA YALT/2HA ,2Hl ,2Ht ,2Hi ,2Ht ,2Hu ,2Hd ,2He /
      DATA Y/128.,128.,128.,368./
      DATA XX/50.,75.,150.,175.,200.,250.,275.,300.,350.,360./
      DATA YY/50.,100.,150.,175.,200.,300.,325.,350.,360.,368./
      DATA IDATE/2H  ,2H D,2Hec,2H 2,2H9.,2H 1,2H96,2H4 /
      DATA YLAB/2H 2,2H 4,2H 6,2H 8,2H10,2H12,2H14,2H16,2H18,2H20/
C  ** INITIALIZE LU DEVICE
      CALL RMPAR(IPAR)
      LU = IPAR(5)
C ** THIS PROGRAM IS TO TEST %PLIB LIBRARY OF SUBROUTINES
C ** WRITTEN FOR THE PLASMA SCOPE
C ** CALL GRAF TO INITIATE PLASMA SCOPE FOR GRAPHING
      CALL GRAF(0)
C ** CALL CLEAR TO CLEAR PLASMA SCOPE
1     CALL CLEAR
C ** CALL SETOR(XORG,YORG) TO INITIALIZE X,Y ORIGIN
      IFG = 0
      CALL SETOR(0.,0.)
C ** CALL SETSC(XSCAL,YSCAL) TO SET SCALE FACTORS
      CALL SETSC(1.,1.)
C ** CALL LINE(X,Y,NXY,MODE) TO PLOT POINTS
C ** X AND Y = THE X,Y CO-ORDINATES
```

```
C ** NXY = THE NUMBER OF POINTS TO BE PLOTTED
C ** MODE = 0 SPECIFIES A WRITE, = 1 SPECIFIES AN ERASE
C ** CALL POINT(X,Y,NXY,MODE) SAME AS ABOVE EXCEPT PLOTS A LINE
      XSHF = -64.
71    CALL LINE(X,Y,2,0)
      DO 70 J=1,6
      I = (J-1)*2 + 1
      CALL LINE(XTIC(I),XTIC1,2,0)
      CALL CHAR(XNUM(I),110.,0,IXNUM(J),2,0,0)
70    CONTINUE
      IF(IFG.GT.0) GO TO 52
      CALL LINE(X(3),Y(3),2,0)
      YLB = Y(1) - 8.
      YNM = Y(1)
      XLB = X(1) - 24.
      DO 10 I=1,10
      YLB = YLB  + 24.
      YNM = YNM + 24.
      J = (I-1)*2 + 1
      CALL CHAR(XLB ,YLB,0,YLAB(I),2,0,0)
      CALL LINE(YTIC1,YTIC(J),2,0)
10    CONTINUE
C ** PRINT SCALAR WIND LEGEND
      CALL CHAR(175.,40.,0,IXLEG,18,0,0)
C ** PRINT ALTITUDE LEGEND
      YAL = 360.
      DO 20 I =1,8
      YAL = YAL - 24.
      CALL CHAR(8.  ,YAL,0,YALT(I),2,0,0)
20    CONTINUE
C ** PRINT HEADER
      CALL CHAR(100.,470.,0,IHEAD,40,0,0)
      CALL CHAR(200.,445.,0,IDATE,16,0,0)
C ** RESET ORIGIN TO PLOT LINE
52    CONTINUE
      CALL SETOR(XSHF,-128.)
C     CALL SETSC(3.2,1.)
      CALL SETSC(1.,1.)
C     CALL LINE(XX,YY,10,0)
      LL = 0.
      DO 40 L=1,800
      LL     = LL + .3
      YY(L) = LL
40    CONTINUE
      IK = 1
      DO 30 K=1,100
      KK = K*8
      READ(8,*) (WS1(IJ),IJ=IK,KK)
      IK = IK +8
30    CONTINUE
```

```
      CALL FILTR(WS1)
      N = 0
      DO 50 L=1,797,4
      IF(WS1(L).GT.100.) GO TO 50
      N = N + 1
      WS1(N) = WS1(L)*2.
      YY(N) = YY(L)
C     CALL POINT(WS1(L),YY(L),1,0)
50    CONTINUE
      CALL LINE(WS1,YY,N,0)
      NJ = (IFG*3) + 1
      R = YY(N) + 4.
      CALL CHAR(WS1(N),R,0,ITIME(NJ),6,0,0)
C     IF(IFG.EQ.0) CALL SETOR(-124.,-110.)
C     IF(IFG.EQ.1) CALL SETOR(-184.,-92.)
C     X(1) = X(1) + 60.
C     X(2) = X(2) + 60.
C     Y(1) = Y(1) - 22.
C     Y(2) = Y(2) - 22.
      IF(IFG.EQ.0) CALL SETOR(-60.,22.)
      IF(IFG.EQ.1) CALL SETOR(-120.,44.)
      IF(IFG.EQ.2) CALL SETOR(-180.,0.)
      IF(IFG.EQ.3) CALL SETOR(-240.,22.)
      IF(IFG.EQ.4) CALL SETOR(-300.,44.)
      IF(IFG.GE.5) GO TO 51
      IFG = IFG + 1 .
      XSHF = XSHF - 60.
      GO TO 71
51    CONTINUE
C     CALL LINE(WS1,YY,800,0)
C   ** CALL TOUCH TO SEE IF USER DESIRES TO CONTINUE OR TERMINATE
      CALL NGRAF
      CALL GRAF(0)
345   CALL DREAD(JIMF1,2,ILINE)
      CALL CHAR(8.0,16.,0,ILINE,64,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.14)GO TO 1
      CALL CHAR(8.,16.,0,IERS,64,0,0)
      CALL DREAD(JIMF1,3,ILINE)
      CALL CHAR(8.,16.,0,ILINE,64,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
      IX = IX/2
      IY = IY/2
      IF(IX.LT.8) GO TO 344
      REWIND 8
      CALL NGRAF
      CALL EXEC(9,JWSPL)
      CALL GRAF(0)
      GO TO 345
C ** CALL NGRAF TO RE-ESTABLISH PLASMA SCOPE FOR TOUCH MODE
344   CALL CLEAR
      CALL NGRAF
C  ** REWIND TAPE
      REWIND 8
      STOP
      END
      SUBROUTINE FILTR(WS1)
      DIMENSION WS1(1)
      DO 1000 IC1=1,798
      IC2 = IC1 + 1
      IC3 = IC1 + 2
      IF(WS1(IC1) .GE. 100.0)GO TO 1000
      DIF1 = WS1(IC1) - WS1(IC2)
      DIF2 = WS1(IC1) - WS1(IC3)
      DIF3 = WS1(IC2) - WS1(IC3)
      IF(ABS(DIF1).GT.1.0 .AND. ABS(DIF3).GT.1.0)WS1(IC2) = WS1(IC1)
      IF((ABS(DIF1).GT.1.0) .AND. (ABS(DIF2).GT.1.0) .AND.
     *   (((DIF1.GT.0.0) .AND. (DIF3.LT.0.0)) .OR.
     *     ((DIF1.LT.0.0) .AND. (DIF3.GT.0.0))))WS1(IC2) = WS1(IC1)
 1000 CONTINUE
      RETURN
C     END
C     SUBROUTINE CLEAR
C     INTEGER RSFF
C     DATA RSFF/017014B/
C     CALL EXEC(2,107B,RSFF,-2)
C     RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
      DIMENSION NAMEF(3),IDCB(276),IBUF(40),ILINE(32)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
10    CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
100   FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
```

```
10      IBUF(I) = IBLK
        RETURN
        END
        END$
```

Program SKEW T (Version I)

```
FTN4
      PROGRAM SKEWT
      DIMENSION IALT(31,3),IDIR(31,3),SPEED(31,3),TEMP(31,3),PRESS(31,3)
      DIMENSION SURDEN(6),V(31),ISWR(20),DPTEMP(31,3),PTEMP(31,3)
      DIMENSION Q(30),A(23),C(23),B(23),E(15),F(4,4),G(4,4),K(23)
      DIMENSION W(31,5),X(40,5),LHEAD(40,80),ALT(31,3)
C     CALL DATE
      P9=3.14159
      N=0
      ISWR(1)=0
      ISWR(2)=0
      ISWR(3)=0
      ISWR(4)=0
      ISWR(5)=0
      ISWR(6)=0
      ISWR(7)=0
      ISWR(8)=0
      ISWR(9)=0
      ISWR(10)=0
      ISWR(11)=0
      ISWR(12)=0
      ISWR(15)=1
61    IUNIT=5
      IF (ISWR(1) .EQ. 0)      IUNIT=1
88    A(1)=0
      N=1
      IF (ISWR(8) .EQ. 1) GO TO 140
      WRITE (6,9010)
C   : DEFINITION OF TERMS:
C TEMP(IALT,N)--TEMPERATURE;PRESS(IALT,N)--PRESSURE; DPTEMP(IALT,N)--HUMIDITY
C
C **LOAD DATA**
      ITIMES=5
      CALL IOHED(LHEAD,ITIMES)
C     READ (8,*) N
      READ (8,9860) LTIM,LDAY,LMON,LM,LYEAR
      ITIMES=19
      CALL IOHED(LHEAD,ITIMES)
C
      WRITE (6,9015)
      READ (1,9850) IFNO
      CALL PTAPE (8,IFNO,0)
C
      T2=9999
      ITIMES=3
      CALL IOHED(LHEAD,ITIMES)
8000  READ (8,9865) ISTIM,ISDAY,ISMON,ISM,ISYEAR
      IF(ISDAY.EQ.0) GO TO 9000
      WRITE(12) -1,1,1000,9000
      WRITE(12,8040)175,0,0,200,ISTIM,ISDAY,ISMON,ISM,ISYEAR
```

```
8040  FORMAT(4I5,20HRAWINSONDE SOUNDING:,I5,1HZ,2X,I2,1X,2A2,I4)
      ITIMES=2
      CALL IOHED(LHEAD,ITIMES)
      READ (8,9870) IALT(1,N),IDIR(1,N),SPEED(1,N),TEMP(1,N),
     1DPTEMP(1,N),PRESS(1,N),SURDEN(N)
      DO 120 I=2,30
115   READ (8,9875) IALT(I,N),IDIR(I,N),SPEED(I,N),TEMP(I,N),
     1DPTEMP(I,N),PRESS(I,N)
      CALL EXEC(13,8,IEQT)
      IEQT=IAND(IEQT,200B)
      IF (IEQT .GT. 0) GO TO 140
      IF (IALT(I,N) .LT. 10) GO TO 115
      IF (IALT(I,N) .GT. 10000) GO TO 115
      JARAY=I
120   CONTINUE
C
C
140   IF (IUNIT .EQ. 2 .OR. IUNIT .EQ. 3) WRITE(6,4020)
4020  FORMAT ("NEED JUMP")
C :N---DATA SET NUMBER; LMON,LDAY,LYEAR,LTIM---LAUNCH DATE/TIME
C :
C : ISTIM--SOUNDING TIME; T2--PREDICTION TIME
C  CONVERTING SOUNDING TIME FROM ZULU TO EDT - AM , PM
      ISTIM=ISTIM-400
      IF (ISTIM .GT. 0) GO TO 250
      ISTIM=2400-ABS(ISTIM)
      ISDAY=ISDAY-1
250   IF (ISTIM .GE. 1300) GO TO 260
      IF (ISTIM .GE. 1200 .AND. ISTIM .LT.1300) GO TO 270
      GO TO 280
260   ISTIM=ISTIM-1200
270   CONTINUE
280   CONTINUE
C   : SURDEN(N)=SURFACE DENSITY
C
C  CONVERT DATA TO METRIC, SORT DATA BY IALT, CAL POT TEMP=PTEMP(IALT,N)
      DO 590 I=1,JARAY
C   :             .........ENGLISH TO METRIC...............
      ALT(I,N)=IALT(I,N)
      ALT(I,N)=.3048*ALT(I,N)
      SPEED(I,N)=.515*SPEED(I,N)
C   :             ..........SORT...............................
509   L=I
510   IF (L .EQ. 1) GO TO 590
      IF (ABS(ALT(L,N)) .GT. ALT(L-1,N)) GO TO 590
      ALT(31,N)=ALT(L-1,N)
      IDIR(31,N)=IDIR(L-1,N)
      SPEED(31,N)=SPEED(L-1,N)
      TEMP(31,N)=TEMP(L-1,N)
      PRESS(31,N)=PRESS(L-1,N)
      DPTEMP(31,N)=DPTEMP(L-1,N)
      ALT(L-1,N)=ALT(L,N)
      IDIR(L-1,N)=IDIR(L,N)
      SPEED(L-1,N)=SPEED(L,N)
      TEMP(L-1,N)=TEMP(L,N)
      PRESS(L-1,N)=PRESS(L,N)
      DPTEMP(L-1,N)=DPTEMP(L,N)
      ALT(L,N)=ALT(31,N)
      IDIR(L,N)=IDIR(31,N)
      SPEED(L,N)=SPEED(31,N)
      TEMP(L,N)=TEMP(31,N)
      PRESS(L,N)=PRESS(31,N)
      DPTEMP(L,N)=DPTEMP(31,N)
      L=L-1
570   GOTO 510
590   CONTINUE
C     .......CALCULATE POTENTIAL TEMPERATURE (DEG K)   PTEMP(ALT,N).....
      DO 690 I=1,JARAY
C     ALT(I,N)=ABS(ALT(I,N))
      PTEMP(I,N)=(TEMP(I,N)+273.15)*((1000/PRESS(I,N))**.288)
690   CONTINUE
C     ........PRINT METEOROLOGICAL DATA..................
725   J=J9
C     IF (ISUR(12) .EQ. 0) WRITE (6,*) "CTR PRINT"
C     IF (ISUR(12) .EQ. 0) WAIT (15000)
      WRITE (6,9220)
      WRITE (6,9140)
      WRITE (6,9140)
      WRITE (6,9140)
      IF (ISUR(15) .EQ. 0) WRITE (6,9230)
      IF (ISUR(15) .EQ. 1) WRITE (6,9240)
      WRITE (6,9250) LTIM, LDAY, LMON, LM, LYEAR
      WRITE (6,9140)
      WRITE (6,9260) ISTIM, ISDAY, ISMON, ISM, ISYEAR
      WRITE (6,9270) T2
      E(6)=.66355
      WRITE (6,9280) N, SURDEN(N)
      WRITE (6,9140)
      WRITE (6,9290)
      WRITE (6,9300)
      DO 850 I=1,JARAY
      SPEED(I,N)=INT(SPEED(I,N)*10)/10
      IALTF=ALT(I,N)/.3048+.5
      IALT(I,N)=ALT(I,N)+.5
      APTEMP=PTEMP(I,N)-273.15
      WRITE (6,9310) I, IALTF, IALT(I,N), IDIR(I,N), SPEED(I,N), TEMP(I,
     1N), APTEMP,DPTEMP(I,N),PRESS(I,N)
850   CONTINUE
C
C....PLOT SKEW T , LOG P DIAGRAM....................................
```

```
C.....SET AT(-1 F,1050MB) & (^82.5 F,500MB).........................
C
      IPEN=-1
      DO 900 I=1,30
      IF(PRESS(I,N).LT.545) GO TO 900
      IX=142.8*(TEMP(I,N)+273.15)-10831.*ALOGT(PRESS(I,N))-3668.
      IY=-34623.*ALOGT(PRESS(I,N))+104603.
      WRITE(12) IPEN,1,IX,IY
      IPEN=1
900   CONTINUE
      WRITE(12) -1,-1,100,-150
      WRITE (12,9031) 100,0,0,125
9031  FORMAT(4I5,7HAMBIENT)
      PAUSE
      IPEN=-1
      DO 925 I=1,30
      IF(PRESS(I,N).LT.545) GO TO 925
      IX=142.8*(DPTEMP(I,N)+273.15)-10831.*ALOGT(PRESS(I,N))-3668.
      IY=-34623.*ALOGT(PRESS(I,N))+104603.
      WRITE(12) IPEN,1,IX,IY
      IPEN=1
925   CONTINUE
      WRITE (12,9032) 100,0,0,125
9032  FORMAT(4I5,9HDEW POINT)
950   WRITE(12) -1,1,9999,9999
      DELP=1000
      DO 975 I=1,30
      IF(PRESS(I,N).LT.545) GO TO 975
      IF(PRESS(I,N).NE.DELP) GO TO 975
      DELP=DELP-50
      IZ=3.28084*IALT(I,N)
      IY=-34623.*ALOGT(PRESS(I,N))+104603.
      WRITE(12)-1,1,899,IY
      WRITE(12,9030) 75,0,0,100,IZ
9030  FORMAT(4I5,I5,3H FT)
      WRITE(12)-1,1,850,IY
      WRITE(12) 1,1,900,IY
      WRITE(12)-1,1,875,IY
      WRITE(12,9020) 75,0,0,100,IALT(I,N)
9020  FORMAT(4I5,I5,7H METERS)
975   CONTINUE
      WRITE(12) -1, 1,250,8500
      WRITE(12,9021) 100,0,0,125
9021  FORMAT(4I5,8HALTITUDE)
      WRITE(12) -1,1,9999,9999
9010  FORMAT ("DATA NUMBER&","&","&" )
9015  FORMAT ("E&aBENTER FILE NUMBER IN 2 DIGIT I FORMAT&E&d@")
9140  FORMAT (70X)
9190  FORMAT ("=======================================================")
9220  FORMAT ("+++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

```
9230  FORMAT (7X,"SPACE SHUTTLE LAUNCH FROM KSC")
9240  FORMAT ("TITAN IIIC LAUNCH FROM KSC")
9250  FORMAT ("␛&dCLaunch time:",I8,9X,"Date:",I2,1X,2A2,1X,I4)
9260  FORMAT ("  TIME OF SOUNDING: ",I8,4X,I2,1X,2A2,1X,I4)
9270  FORMAT ("  TIME OF PREDICTION: ",I4)
9280  FORMAT ("DATA SET: ",I2,13X,"SURFACE DENSITY(GM/M**3)",F8.2)
9290  FORMAT ("␛&dBLAYER   ALTITUDE    DIRECTION SPEED  TEMP POT-TEMP
     1␛&dB  D P TEMP  PRESSURE")
9300  FORMAT ("␛&dN No. (FEET) (METERS) (DEGREES) (M SEC) (DEGREE CENTIG
     1␛&dNRADE) (MILLIBARS)")
9310  FORMAT (I2,I7,2X,I5,7X,13,4X,F4.1,4X,F4.1,2X,F5.2,2X,F4.1,
     12X,F7.2)
9850  FORMAT (I2)
9860  FORMAT (26X,I4,6X,I2,1X,2A2,I4)
9865  FORMAT (I4,3X,I2,1X,2A2,I4)
9870  FORMAT (I6,1X,2F4.1,F6.1,F6.1,F7.2,11X,F7.2)
9875  FORMAT (I6,1X,2F4.1,F6.1,F6.1,F7.2)
      REWIND 8
9999  END
      SUBROUTINE IOHED(LHEAD,ITIMES)
      DIMENSION LHEAD(40,80)
      DO 110 I=1,ITIMES
      READ (8,9855) (LHEAD(I,J),J=1,40)
      WRITE (6,9855) (LHEAD(I,J),J=1,40)
110   CONTINUE
9855  FORMAT (40A2)
      RETURN
      END
      END$
```

Program SKEW T (Version II)

```
FTN4
      PROGRAM SKEW
      DIMENSION IALT(31,3),IDIR(31,3),SPEED(31,3),TEMP(31,3),PRESS(31,3)
      DIMENSION SURDEN(6),V(31),ISUR(20),DPTEMP(31,3),PTEMP(31,3)
      DIMENSION Q(30),A(23),C(23),B(23),E(15),F(4,4),G(4,4),K(23)
      DIMENSION W(31,5),X(40,5),LHEAD(40),ALT(31,3),IDCB(256)
      DIMENSION IBUF(80), NAME(3)
      DATA NAME/2HTE,2HMP,2HRD/
C     IFMT = 0  DATA IS VAR PT FORMAT
C     IFMT = 1  DATA IS FIX PT FORMAT
C     IREAD =0  DATA IS ON DISC
C     IREAD =1  DATA IS ON TAPE
C  ** CONVERT RH AND TEMP TO DEWPT **
      SFY = 10./9.
      C = 8.42926604
      D = 1.82717843
      E = 0.071208271
C  ** INITIALIZE DATA FORMATS AND INPUT UNITS **
      IFMT = 0
      IREAD = 0
      WRITE(1,11)
11    FORMAT("E&dB  ENTER CASE NUMBER  ")
      READ(1,20) N
      WRITE(1,10)
10    FORMAT("E&dB  ENTER NUMBER OF POINTS  ")
      READ(1,20)  NN
20    FORMAT(I2)
      WRITE(1,15)
15    FORMAT("E&dB  ENTER  ISTIM ISDAY ISMON ISM ISYEAR  ")
      READ(1,16) ISTIM,ISDAY,ISMON,ISM,ISYEAR
16    FORMAT(I4,I2,2A2,I4)
C **  INITIALIZE DATES **
      LTIM = 1500
      LDAY = 29
      LMON = 2HOC
      LM   = 2HT
      LYEAR = 1975
      IF(IREAD.EQ.0) CALL OPEN(IDCB,IERR,NAME,0)
C     CALL DATE
      P9=3.14159
      M=0
      ISUR(1)=0
      ISUR(2)=0
      ISUR(3)=0
      ISUR(4)=0
      ISUR(5)=0
      ISUR(6)=0
      ISUR(7)=0
      ISUR(8)=0
      ISUR(9)=0
      ISWR(10)=0
      ISWR(11)=0
      ISWR(12)=0
      ISWR(15)=1
61    IUNIT=5
      IF (ISWR(1) .EQ. 0)     IUNIT=1
88    A(1)=0
      IF (ISWR(8) .EQ. 1) GO TO 140
      IF(IFMT.EQ.1) WRITE(6,9010)
C   · DEFINITION OF TERMS:
C TEMP(IALT,N)--TEMPERATURE;PRESS(IALT,N)--PRESSURE; DPTEMP(IALT,N)--HUMIDIT
C
C **  LOAD DATA FROM TAPE **
      IF(IREAD.EQ.0) GO TO 121
      ITIMES=5
      CALL IOHED(LHEAD,ITIMES)
C     READ (8,*) N
      READ (8,9860) LTIM,LDAY,LMON,LM,LYEAR
      ITIMES=19
      CALL IOHED(LHEAD,ITIMES)
C
      WRITE (6,9015)
      READ (1,9850) IFNO
      CALL PTAPE (8,IFNO,0)
C
      T2=9999
      ITIMES=3
      CALL IOHED(LHEAD,ITIMES)
8000  READ (8,9865) ISTIM,ISDAY,ISMON,ISM,ISYEAR
      IF(ISDAY.EQ.0) GO TO 8000
      WRITE(12) -1,1,1000,9750
      WRITE(12,8040)175,0,0,200,ISTIM,ISDAY,ISMON,ISM,ISYEAR
8040  FORMAT(4I5,20HRAWINSONDE SOUNDING:,I5,1HZ,2X,I2,1X,2A2,I4)
      ITIMES=2
      CALL IOHED(LHEAD,ITIMES)
      READ (8,9870) IALT(1,N),IDIR(1,N),SPEED(1,N),TEMP(1,N),
     1DPTEMP(1,N),PRESS(1,N),SURDEN(N)
      DO 120 I=2,30
115   READ (8,9875) IALT(I,N),IDIR(I,N),SPEED(I,N),TEMP(I,N),
     1DPTEMP(I,N),PRESS(I,N)
      CALL EXEC(13,8,IEQT)
      IEQT=IAND(IEQT,200B)
      IF (IEQT .GT. 0) GO TO 140
      IF (IALT(I,N) .LT. 10) GO TO 115
      IF (IALT(I,N) .GT. 10000) GO TO 115
      JARAY=I
120   CONTINUE
      IF(IREAD.EQ.1) GO TO [illegible]
C     **LOAD DATA FROM DISC ***
C
```

```
121    CALL READF(IDCB,IERR,IBUF)
       CALL CODE
       READ(IBUF,301)
       T2 = 9999
       CALL READF(IDCB,IERR,IBUF)
       CALL CODE
       READ(IBUF,302) SURDEN(N)
       WRITE(12) -1,1,1000,9750
       WRITE(12,8040) 175.0,0,200,ISTIM,ISDAY,ISMON,ISM,ISYEAR
       DO 122 I =1,NN
123    CALL READF(IDCB,IERR,IBUF)
       CALL CODE
       READ(IBUF,303) IALT(I,N),IDIR(I,N),SPEED(I,N),TEMP(I,N),
      1PRESS(I,N),DPTEMP(I,N)
       IF(IALT(I,N).LT.1)  GO TO 123
       IF(IALT(I,N).GT.10000) GO TO 123
       T = 1000./(TEMP(I,N) + 273.15)
       E1 = (DPTEMP(I,N)/100.)*10.**(C - D*T - E*T*T)
       C1 = ALOGT(E1) - C
       DT = (SQRT(D*D - 4*E*C1) - D)/(2*E)
       DT = (1000./DT) - 273.15
       DPTEMP(I,N) = DT
       JARAY = I
122    CONTINUE
C
140    IF (IUNIT .EQ. 2 .OR. IUNIT .EQ. 3) WRITE(6,4020)
4020   FORMAT ("NEED JUMP")
C :N---DATA SET NUMBER; LMON,LDAY,LYEAR,LTIM---LAUNCH DATE/TIME
C :
C : ISTIM--SOUNDING TIME; T2--PREDICTION TIME
C  CONVERTING SOUNDING TIME FROM ZULU TO EDT - AM , PM
       ISTIM=ISTIM-400
       IF(IFMT.EQ.0) ISTIM = ISTIM-200
       IF (ISTIM .GT. 0) GO TO 250
       ISTIM=2400-ABS(ISTIM)
       ISDAY=ISDAY-1
250    IF (ISTIM .GE. 1300) GO TO 260
       IF (ISTIM .GE. 1200 .AND. ISTIM .LT.1300) GO TO 270
       GO TO 280
260    ISTIM=ISTIM-1200
270    CONTINUE
280    CONTINUE
C   : SURDEN(N)=SURFACE DENSITY
C
C  CONVERT DATA TO METRIC, SORT DATA BY IALT, CAL POT TEMP=PTEMP(IALT,N)
       DO 590 I=1,JARAY
C   :             .........ENGLISH TO METRIC...............
       ALT(I,N)=IALT(I,N)
       IF(IFMT.EQ.0) GO TO 509
       ALT(I,N)=.3048*ALT(I,N)
      SPEED(I,N)=.515*SPEED(I,N)
C  :            ...........SORT.......................................
509   L=I
510   IF (L .EQ. 1) GO TO 590
      IF (ABS(ALT(L,N)) .GT. ALT(L-1,N)) GO TO 590
      ALT(31,N)=ALT(L-1,N)
      IDIR(31,N)=IDIR(L-1,N)
      SPEED(31,N)=SPEED(L-1,N)
      TEMP(31,N)=TEMP(L-1,N)
      PRESS(31,N)=PRESS(L-1,N)
      DPTEMP(31,N)=DPTEMP(L-1,N)
      ALT(L-1,N)=ALT(L,N)
      IDIR(L-1,N)=IDIR(L,N)
      SPEED(L-1,N)=SPEED(L,N)
      TEMP(L-1,N)=TEMP(L,N)
      PRESS(L-1,N)=PRESS(L,N)
      DPTEMP(L-1,N)=DPTEMP(L,N)
      ALT(L,N)=ALT(31,N)
      IDIR(L,N)=IDIR(31,N)
      SPEED(L,N)=SPEED(31,N)
      TEMP(L,N)=TEMP(31,N)
      PRESS(L,N)=PRESS(31,N)
      DPTEMP(L,N)=DPTEMP(31,N)
      L=L-1
570   GOTO 510
590   CONTINUE
C  ........CALCULATE POTENTIAL TEMPERATURE (DEG K)   PTEMP(ALT,N).....
      DO 690 I=1,JARAY
C     ALT(I,N)=ABS(ALT(I,N))
      PTEMP(I,N)=(TEMP(I,N)+273.15)*((1000/PRESS(I,N))**.288)
690   CONTINUE
C  .........PRINT METEOROLOGICAL DATA...................
725   J=J9
C     IF (ISWR(12) .EQ. 0) WRITE (6,*) 'CTR PRINT'
C     IF (ISWR(12) .EQ. 0) WAIT (15000)
      IF(IFMT.EQ.0) GO TO 727
      WRITE (6,9220)
      WRITE (6,9140)
      WRITE (6,9140)
      WRITE (6,9140)
      IF (ISWR(15) .EQ. 0) WRITE (6,9230)
      IF (ISWR(15) .EQ. 1) WRITE (6,9240)
727   IF(IFMT.EQ.0) WRITE(6,9333)
      IF(IFMT.EQ.0) WRITE(6,9334)
      IF(IFMT.EQ.0) WRITE(6,9335) LTIM,LDAY,LMON,LM,LYEAR
      IF(IFMT.EQ.0) GO TO 728
      WRITE (6,9250) LTIM, LDAY, LMON, LM, LYEAR
      WRITE (6,9140)
728   WRITE (6,9260) ISTIM, ISDAY, ISMON, ISM, ISYEAR
      WRITE (6,9270) T2
      E(6)=.66355
      WRITE (6,9280) N, SURDEN(N)
      WRITE (6,9140)
      WRITE (6,9290)
      WRITE (6,9300)
      DO 850 I=1,JARAY
      SPEED(I,N)=INT(SPEED(I,N)*10)/10
      IALTF=ALT(I,N)/.3048+.5
      IALT(I,N)=ALT(I,N)+.5
      APTEMP=PTEMP(I,N)-273.15
      WRITE (6,9310) I, IALTF, IALT(I,N), IDIR(I,N), SPEED(I,N), TEMP(I,
     1N), APTEMP,DPTEMP(I,N),PRESS(I,N)
850   CONTINUE
C
C.....PLOT SKEW T , LOG P DIAGRAM......................................
C **  COMPUTE IY USING V1,V2,V3 FOR LOGRITHMIC PLOT CONVERSIONS **
      V1 = 58470.457
      V2 = -28656.688
      V3 = 3078.846
C.....SET AT(-1 F,1050MB) & (^82.5 F,500MB)............................
C
      IPEN=-1
      DO 900 I=1,NN
      IF(PRESS(I,N).LT.545) GO TO 900
      IX=142.8*(TEMP(I,N)+273.15)-10831.*ALOGT(PRESS(I,N))-3668.
      IY= V1*ALOGT(PRESS(I,N)) + V2*(ALOGT(PRESS(I,N)))**2 +
     1V3*(ALOGT(PRESS(I,N)))**3
      IY = IY*SFY
      WRITE(12) IPEN,1,IX,IY
      IPEN=1
900   CONTINUE
      WRITE(12) -1,-1,100,-150
      WRITE (12,9031) 100,0,0,125
9031  FORMAT(4I5,7HAMBIENT)
C     PAUSE
      IPEN=-1
      DO 925 I=1,NN
      IF(PRESS(I,N).LT.545) GO TO 925
      IX=142.8*(DPTEMP(I,N)+273.15)-10831.*ALOGT(PRESS(I,N))-3668.
      IY=V1*ALOGT(PRESS(I,N)) + V2*(ALOGT(PRESS(I,N)))**2 +
     1V3*(ALOGT(PRESS(I,N)))**3
      IY = IY*SFY
      WRITE(12) IPEN,1,IX,IY
      IPEN=1
925   CONTINUE
      WRITE(12) -1,-1,100,-150
      WRITE (12,9032) 100,0,0,125
9032  FORMAT(4I5,9HDEW POINT)
950   WRITE(12) -1,1,9999,9999
      DELP=850
      DO 975 I=1,NN
      IF(PRESS(I,N).LT.545) GO TO 975
      IF(PRESS(I,N).NE.DELP) GO TO 975
      DELP=DELP-50
      IZ=3.28084*IALT(I,N)
      IY=V1*ALOGT(PRESS(I,N)) + V2*(ALOGT(PRESS(I,N)))**2 +
     1V3*(ALOGT(PRESS(I,N)))**3
      IY = IY*SFY
      WRITE(12)-1,1,200,IY
      WRITE(12,9030) 75,0,0,100,IZ
9030  FORMAT(4I5,I5,3H FT)
      WRITE(12)-1,1,850,IY
      WRITE(12) 1,1,900,IY
      WRITE(12)-1,1,875,IY
      WRITE(12,9020) 75,0,0,100,IALT(I,N)
9020  FORMAT(4I5,I5,7H METERS)
303   FORMAT(5X,I4,7X,I3,7X,F5.1,4X,F5.1,6X,F5.1,5X,F5.1,19X)
301   FORMAT(80X)
302   FORMAT(59X,F6.1,15X)
975   CONTINUE
      WRITE(12) -1, 1,250,9500
      WRITE(12,9021) 100,0,0,125
9021  FORMAT(4I5,8HALTITUDE)
      WRITE(12) -1,1,9999,9999
9010  FORMAT (1X,"DATA NUMBER▲","▲","▲" )
9015  FORMAT (1X,"Ɛ&dBENTER FILE NUMBER IN 2 DIGIT I FORMAT▲Ɛ&d@")
9140  FORMAT (70X)
9190  FORMAT (1X,"================================================")
9220  FORMAT (1X,"++++++++++++++++++++++++++++++++++++++++++++++++")
9230  FORMAT (7X,"SPACE SHUTTLE LAUNCH FROM KSC")
9240  FORMAT (1X,"TITAN IIIC LAUNCH FROM KSC")
9250  FORMAT (1X,"Ɛ&aCLaunch time:",I8,9X,"Date:",I2,1X,2A2,1X,I4)
9260  FORMAT (1X,"  TIME OF SOUNDING: ",I8,4X,I2,1X,2A2,1X,I4)
9270  FORMAT (1X,"  TIME OF PREDICTION: ",I4)
9280  FORMAT (1X,"DATA SET: ",I2,13X,"SURFACE DENSITY(GM/M**3)",F8.2)
9290  FORMAT (5X,"LAYER ALTITUDE",2X,"DIRECTION",1X,"SPEED",3X,
     1"TEMP",1X,"DP-TEMP",1X,"PRESSURE")
9300  FORMAT(1X,"NO.",2X,"FEET",1X,"METERS",3X,"DEGREES",3X,"M SEC",
     14X,"DEGREE CENTIGRADE",1X,"MILLIBARS")
9333  FORMAT(1X,"*****************************************")
9334  FORMAT(1X,"SPACE SHUTTLE SRM DDTE PROGRAM TEST FIRINGS AT THIOKOL
     1 WASATCH")
9335  FORMAT(1X,"      LAUNCH TIME:  ",I8,9X,"DATE:",I2,1X,2A2,1X,I4)
9310  FORMAT (1X,I2,I7,2X,I5,7X,I3,4X,F4.1,4X,F4.1,2X,F5.2,2X,F4.1,
     12X,F7.2)
9850  FORMAT (I2)
9860  FORMAT (26X,I4,6X,I2,1X,2A2,I4)
9865  FORMAT (I4,3X,I2,1X,2A2,I4)
9870  FORMAT (I6,1X,2F4.1,F6.1,F6.1,F7.2,11X,F7.2)
9875  FORMAT (I6,1X,2F4.1,F6.1,F6.1,F7.2)
```

```
      IF(IFMT.EQ.1) REWIND 8
      IF(IFMT.EQ.0) CALL CLOSE(IDCB)
9999  END
      SUBROUTINE IOHED(LHEAD,ITIMES)
      DIMENSION LHEAD(40)
      DO 110 I=1,ITIMES
      READ (8,9855)(LHEAD(N),N=1,40)
      WRITE (6,9856)(LHEAD(N),N=1,40)
110   CONTINUE
9855  FORMAT (40A2)
9856  FORMAT (1X,40A2)
      RETURN
      END
      END$
```

Program PUFF

```
FTN4,L
      PROGRAM PUFF
C MAIN
      DIMENSION ICARDS(40), IBUF(40), IDCB(256), NAME(3)
      REAL MWE,MEDOT,MWINF,ME,MINF,M1,M2,LT
      COMMON V(11),DV(11),T,DT,NV,G(3),RAD,PI,R,UGC,MEDOT,TOFF,TC,
     1ME,MINF,VC,RHOE,CPE,RE,GAME,MWE,TE,PE,UEX,UEY,UEZ,UE,
     2HC,CD,RHOINF,CPINF,RINF,GAMINF,MWINF,TINF,PINF,THETA,
     3GAMMA,PC,APPP,VELW(3),H,CR,LT,XC,ACS,ASP,IFLAG

      DATA  DTO/0.01/, IPR,J,IOFF/3*0/
      DATA NAME/2H&P,2HUF,2HFD/
C     DATA V,DV,T,DT,NV/23*0.,.01,11/,G,RAD,PI/2*0.,980.7,57.296,3.1416/
C     DATA TE/1000./, TINF/288./,PE/1.0/, PINF/1.0/,GAME/1.2678/
C     DATA GAMINF/1.4/, MWE/19.648/, MWINF/28.966/, UGC/82.0567/, HC/0.0/
C LOAD INITIAL DATA VALUES REPLACING DATA STATEMENTS
      TE=1000.
      TINF=288.
      PE=1.0
      PINF=1.0
      GAME=1.2678
      GAMINF=1.4
      MWE=19.648
      MWINF=28.966
      UGC=82.0567
      HC=0.0
      DO 111 I=1,11
      V(I) = 0.0
      DV(I) = 0.0
111   CONTINUE
      T=0.0
      DT=.01
      NV=11
      G(1) = 0.0
      G(2) = 0.0
      G(3) = 980.7
      RAD= 57.296
      PI = 3.1416
C     CALL ERRSET(208,-1,-1,1)
C ESTABLISH PARAMETERS
C FOLLOWING ARE DEFINITIONS OF INPUT DATA...
C   TOFF...TIME WHEN JET IS SHUT OFF(SEC)
C   TMAX...TIME WHEN SOLUTION IS STOPPED(SEC)
C   DTI....INTEGRATION STEP SIZE(SEC)
C   IPRINT..NUMBER OF STEPS BETWEEN PRINTOUT(IPRINT=1,PRINTS DATA EACH STEP)
C   IFLAG.CONTROLS DEBUG PRINTOUT. IFLAG=1 WRITES FORMAT 100 IN SUBROUTINE
C          DERIV AND FORMATS 100-105 IN SUBROUTINE SHAPE
C   IUNITS.CONTROLS UNITS OF OUTPUT(0=CM + G, 1=M + KG)
C   R......JET EXIT RADIUS(CM)
C   UE.....JET EXIT VELOCITY (CM/SEC)
```

```
C    GAMMA. JET EXIT ELEVATION ANGLE(DEG) HORIZONTAL IS ZERO
C    TE   . JET EXIT TEMPERATURE(DEG-K)
C    GAME   JET EXIT SPECIFIC HEAT RATIO
C    MWE    JET EXIT MOLECULAR WEIGHT
C    TINF   ATMOSPHERIC TEMPERATURE(DEG-K)
C    PINF   ATMOSPHERIC PRESSURE(ATMOSPHERES)
C    GAMINF ATMOSPHERIC SPECIFIC HEAT RATIO
C    MWINF  ATMOSPHERIC MOLECULAR WEIGHT
C    APPP1. ENTRAINMENT COEFFICIENT BEFORE TAIL EXCEEDS 15 DIAMETERS
C    APPP2. ENTRAINMENT COEFFICIENT AFTER  TAIL EXCEEDS 15 DIAMETERS
C      CD1. DRAG COEFFICIENT BEFORE TAIL EXCEEDS 15 DIAMETERS
C      CD2  DRAG COEFFICIENT AFTER TAIL EXCEEDS 15 DIAMETERS
C    THETA  JET EXIT AZIMUTH WITH RESPECT TO X-COORDINATE(DEG)
C    VELW . WIND VECTOR COMPONENTS IN XYZ-COORDINATES(CM/SEC)

C  OPEN INPUT DATA FILE &PUFFD
      CALL OPEN(IDCB,IERR,NAME,0)
C  READ AND PRINT OUT INPUT TEST DATA
      WRITE(6,303)
303   FORMAT(1H1,"INPUT DATA IS AS FOLLOWS: ")
      DO 320 I=1,5
      CALL BLANK(IBUF)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,301) ICARDS
301   FORMAT(40A2)
      WRITE(6,302) (ICARDS(N),N=1,40)
302   FORMAT(1H ,40A2)
320   CONTINUE
C  REWIND AND READ INPUT DATA TO PROCESS
      CALL RWNDF(IDCB,IERR)
      CALL BLANK(IBUF)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,304) TOFF,TMAX,DT1,R,UE
304   FORMAT(5(7X,F8.2))
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,304) GAMMA,TE,GAME,MWE,TINF
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,306) PINF,GAMINF,MWINF,APPP1,APPP2
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,307) CD1, CD2, THETA, VELW(1), VELW(2)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,305) VELW(3), IPRINT,IFLAG,IUNIT
305   FORMAT(7X,F8.0,3(7X,I8))
306   FORMAT(5(7X,F8.5))
```

```
307   FORMAT(5(7X,F8.3))
C     WRITE(6,388) TOFF,TMAX,DTI,R,UE
C     WRITE(6,388) GAMMA,TE,GAME,MUE,TINF
C     WRITE(6,388) PINF,GAMINF,MUINF,APPP1,APPP2
C     WRITE(6,388) CD1,CD2,THETA,VELU(1),VELU(2)
C     WRITE(6,389) VELU(3),IPRINT,IFLAG,IUNITS
C388  FORMAT(1H ,"SAMPLE INPUT",/,5E20.6)
C389  FORMAT(1H ,E20.6,3I10)
      VELU(2) = 0.
      APPP=APPP1
      CD=CD1
C COMPUTE SOME OTHER INVARIANT PARAMETERS
      D15=30.*R
      PE=PINF
      PC=PE
      RE=UGC/MUE
      RINF=UGC/MUINF
      CPE=RE*GAME/(GAME-1.0)
      CPINF=RINF*GAMINF/(GAMINF-1.0)
      RHOE=PE/RE/TE
      RHOINF=PINF/RINF/TINF
      TRAD=THETA/RAD
      GRAD=GAMMA/RAD
      UEX=UE*COS(GRAD)*COS(TRAD)
      UEY=UE*COS(GRAD)*SIN(TRAD)
      UEZ=UE*SIN(GRAD)
      AE=PI*R **2
      MEDOT=RHOE*AE*UE
C ESTABLISH OUTPUT CONSTANTS
      C1=1.0
      C2=1.0
      C3=1.0
      IF (IUNITS.EQ.0)  GO TO 13
      C1=0.01
      C2=1.E-6
      C3=0.001
 13   CONTINUE
C     WRITE(6,200) UGC,MEDOT,RHOE,AE,RHOINF,RE,RINF,CPE,CPINF
 200  FORMAT(" ",9E12.5)
      WRITE (6,210)
C INTEGRATE FOR TMAX SECONDS, PRINT EVERY IPRINT STEPS.
 10   CONTINUE
      DT=0.1
      J=J+1
      IPR=IPR+1
C ALWAYS USE DT=DTO  DURING FIRST 0.1 SEC OF JET ON
      DT=DTI
C     IF (T.GE.0..AND.T.LT.0.099)  DT=DTO
      IF(T.GE.0..AND.T.LT.1.) DT = DTO
      CALL RK4
```

```
C IF TAIL LENGTH GT 15 DIAMETERS, CHANGE ENTRAINMENT COEF
      IF (LT.LT.D15.OR.APPP.EQ.APPP2)  GO TO 11
      APPP=APPP2
      CD=CD2
      WRITE(6,120)
 120  FORMAT("OCLOUD TAIL EXCEEDS 15 DIAMETERS.  ENTRAINMENT COEF INCREA
     1SES")
      WRITE(6,210)
 11   CONTINUE
      IF (T.LT.TOFF.OR.IOFF.EQ.1)  GO TO 12
      WRITE(6,130)  TOFF
130   FORMAT(1H ,"JET SHUT OFF AT T = ",F6.2," SEC")
      WRITE(6,210)
 210  FORMAT("0",3X,"T",6X,"X",6X,"Y",6X,"Z",5X,"VX",5X,"VY",5X,"VZ",5X,
     1  "TC",5X,"XT",5X,"YT",5X,"ZT",5X,"LT",5X,"LS",5X,"CR",
     2 7X,"ME",6X,"MINF",6X,"VOL")
      IOFF=1
 12   CONTINUE
      IF (IPR.LT.IPRINT)  GO TO 10
C WRITE INTEGRATION VARIABLES AND CLOUD DIMENSIONS
      IPR=0
C SET MODE TO 1
      MODE=1

      CALL EVAL1(MODE)
      IF (IFLAG.EQ.0)  GO TO 9
      WRITE(6,210)
 9    CONTINUE
C CONVERT TO METERS, KG IF REQUIRED
      X=V(6)*C1
      Y=V(7)*C1
      Z=V(8)*C1
      VX=DV(6)*C1
      VY=DV(7)*C1
      VZ=DV(8)*C1
      ELT=LT*C1
      XT=V(9)*C1
      YT=V(10)*C1
      ZT=V(11)*C1
      ELS=(LT+H-CR)*C1
      RCLD=CR*C1
      M1=ME*C3
      M2=MINF*C3
      VOL=VC*C2
      WRITE(6,220) T,X,Y,Z,VX,VY,VZ,TC,XT,YT,ZT,ELT,ELS,RCLD,M1,M2,VOL
 220  FORMAT(" ",F5.2,6F7.2,F7.0,6F7.2,3F9.1)
      IF (T.LT.TMAX)  GO TO 10
      CALL CLOSE(IDCB,IERR)
      STOP
      END
      SUBROUTINE BLANK(IBUF)
      DIMENSION IBUF(40)
      DATA IBLK,2H
      DO 66 N=1,40
66    IBUF(N) = IBLK
      RETURN
      END
      SUBROUTINE DERIV
      REAL MWE,MEDOT,MWINF,LT
      COMMON V(11),DV(11),T,DT,NV,G(3),RAD,PI,R,UGC,MEDOT,TOFF,TC,
     1ME,MINF,VC,RHOE,CPE,RE,GAME,MWE,TE,PE,UEX,UEY,UEZ,UE,
     2HC,CD,RHOINF,CPINF,RINF,GAMINF,MWINF,TINF,PINF,THETA,
     3GAMMA,PC,APPP,VELW(3),H,CR,LT,KC,ACS,ASP,IFLAG
      REAL MA,MC,ME,M1X,M1Y,M1Z,M2X,M2Y,M2Z,M3X,M3Y,M3Z,MINF,MOMX,
     1 MOMY,MOMZ,MDTINF
      EQUIVALENCE (V(1),MC),(V(2),MOMX),(V(3),MOMY),(V(4),MOMZ),
     1     (V(5),EC),(V(6),CGX),(V(7),CGY),(V(8),CGZ),(V(9),STX),
     2     (V(10),STY),(V(11),STZ)
      DIMENSION UINF(3)
      DATA DXL,DYL,VCL,SCGL/4*0./
      MODE=0
      IF (T .NE. 0.0) GO TO 10
C     COMPUTE INITIAL DERIVATIVES
      DV(1)=MEDOT
      DV(2)=MEDOT*UEX
      DV(3)=MEDOT*UEY
      DV(4)=MEDOT*UEZ
      DV(5)=MEDOT*CPE*TE
      DV(6)=UE**3
      DV(7)=0
      DV(8)=0
      DV(9)=0
      DV(10)=0
      DV(11)=0
      ME=0
      RETURN
10    CALL EVAL(MODE)
      RETURN
      END
      SUBROUTINE EVAL(MODE)
      COMMON V(11),DV(11),T,DT,NV,G(3),RAD,PI,R,UGC,MEDOT,TOFF,TC,
     1ME,MINF,VC,RHOE,CPE,RE,GAME,MWE,TE,PE,UEX,UEY,UEZ,UE,
     2HC,CD,RHOINF,CPINF,RINF,GAMINF,MWINF,TINF,PINF,THETA,
     3GAMMA,PC,APPP,VELW(3),H,CR,LT,XC,ACS,ASP,IFLAG
      REAL MWE,MEDOT,MWINF,LT
      REAL MA,MC,ME,M1X,M1Y,M1Z,M2X,M2Y,M2Z,M3X,M3Y,M3Z,MINF,MOMX,
     1MOMY,MOMZ,MDTINF
      EQUIVALENCE (V(1),MC),(V(2),MOMX),(V(3),MOMY),(V(4),MOMZ),
     1 (V(5),EC),(V(6),CGX),(V(7),CGY),(V(8),CGZ),(V(9),STX),
     1 (V(10),STY),(V(11),STZ)
```

```
      DIMENSION UINF(3)
      DATA DXL,DYL,VCL,SCGL/4*0./
C THIS ENTRY USED TO FIND CLOUD SHAPE. NOT USED WHEN INTEGRATING.
C     MODE=1
 10   CONTINUE
C AT TOFF SET MEDOT=0 + HOLD ME CONSTANT
      IF (T.GE.TOFF)  GO TO 12
      ME=MEDOT*T
      GO TO 14
 12   CONTINUE
      IF (MEDOT.EQ.0)  GO TO 14
      ME=MEDOT*TOFF
      MEDOT=0.
 14   CONTINUE
      MINF=MC-ME
      CPC=(MINF*CPINF+ME*CPE)/MC
      TC=EC/(MC*CPC)
      RC=(MINF*RINF+ME*RE)/MC
      RHOC=PC/(RC*TC)
      VC=MC/RHOC
      DX=CGX-STX
      DY=CGY-STY
      DZ=CGZ-STZ
      SCG=SQRT(DX*DX+DY*DY+DZ*DZ)
C CALL CLOUD SHAPE SUBROUTINE TO GET ACS
      VZ=VC
      CALL SHAPE(VZ,SCG)
C IF MODE=1, EVALUATE CLOUD SHAPE BUT NO DERRIVATIVES REQUIRED.
      IF (MODE.EQ.1)  GO TO 20
      MA=0.5*RHOINF*VC
      CALL WIND (CGZ,UINF)
      UCX=(MOMX+MA*UINF(1))/(MC+MA)
      UCY=(MOMY+MA*UINF(2))/(MC+MA)
      UCZ=(MOMZ+MA*UINF(3))/(MC+MA)
      UC=SQRT(UCX**2+UCY**2+UCZ**2)
      ELS=(LT+H-CR)/SCG
      CSX=STX+DX*ELS
      CSY=STY+DY*ELS
      CSZ=STZ+DZ*ELS
      ELT=LT/SCG
      SLX=STX+DX*ELT
      SLY=STY+DY*ELT
      SLZ=STZ+DZ*ELT
      SL=SQRT(SLX*SLX+SLY*SLY+SLZ*SLZ)
      ST=SQRT(STX**2+STY**2+STZ**2)
      URX=UINF(1)-UCX
      URY=UINF(2)-UCY
      URZ=UINF(3)-UCZ
      UR=SQRT(URX**2+URY**2+URZ**2)
      MDTINF=RHOINF*UR*ACS*APPP
      M1X=MEDOT*UEX
      M1Y=MEDOT*UEY
      M1Z=MEDOT*UEZ
      CONST=2.0*MA-MC
      M2X=G(1)*CONST
      M2Y=G(2)*CONST
      M2Z=G(3)*CONST
      CONST=UR*ASP*CD*RHOINF
      M3X=URX*CONST
      M3Y=URY*CONST
      M3Z=URZ*CONST
      E1=MEDOT*CPE*TE
      E2=MDTINF*CPINF*TINF
      SBCON=1.355E-12*41.293
      EMISS=0.4
      E3=EMISS*SBCON*ACS*(TINF**4-TC**4)
      E4=2.0*PINF*MEDOT/RHOINF
      DV(1)=MEDOT+MDTINF
      DV(2)=M1X+M2X+M3X
      DV(3)=M1Y+M2Y+M3Y
      DV(4)=M1Z+M2Z+M3Z
      DV(5)=E1+E2+E3
      DV(6)=UCX
      DV(7)=UCY
      DV(8)=UCZ
      IF (T.LT.TOFF)  GO TO 30
      DV(9)=UEX*DX*LT/(STX*SCG+DX*LT+1.0E-9)+UCX
      DV(10)=UEY*DY*LT/(STY*SCG+DY*LT+1.0E-9)+UCY
      DV(11)=UEZ*DZ*LT/(STZ*SCG+DZ*LT+1.0E-9)+UCZ
 30   CONTINUE
      IF (IFLAG.EQ.0)  GO TO 9
      FDELTA=M3X-MA*DV(6)
      VB=VC/RE**3
      XB=XCG/RE
C     WRITE DEBUG OUTPUT
      WRITE(6,100) ME,MINF,CPC,TC,RC,RHOC,VC,MA,UCX,UCY,UCZ,UC,SCG,URX,
     1URY,URZ,UR,M1X,M1Y,M1Z,M2X,M2Y,M2Z,M3X,M3Y,M3Z,E1,E2,E3,E4
     2 ,MDTINF,CR,XC, ASP,H,LT,XB,VB,FDELTA,ACS,T,ST
 9    CONTINUE
 100  FORMAT(1H0,'ME,MINF,CPC,TC,RC,RHOC=',6E15.5/,
     1 '      VC,MA,UCX,UCY,UCZ,UC=',6E15.5/,
     2 '   SCG,URX,URY,URZ,UR,M1X=',6E15.5/,
     3 ' M1Y,M1Z,M2X,M2Y,M2Z,M3X,=',6E15.5/,
     4 '      M3Y,M3Z,E1,E2,E3,E4,=',6E15.5/,
     5 '   MDTINF,CR,XC, ASP,H,LT=',6E15.5/,
     6 '   XB,VB,FDELTA,ACS,T,ST =',6E15.5)
 20   CONTINUE
      RETURN
      END
      SUBROUTINE WIND(H,UINF)
      DIMENSION UINF(3)
      COMMON DUM6(24),IDUM6,DUM7(37),VELU(3),DUM8(6),IDUM8
      DO 100 I=1,3
      UINF(I) = VELU(I)
100   CONTINUE
      RETURN
      END
      SUBROUTINE SHAPE(VZ,CG)
      DIMENSION XCOF(4),COF(4),ROOTR(3),ROOTI(3)
      DOUBLE PRECISION Q,QQ,A,B,Y1,PBY3CU,ROOTQ,XBAR,PBAR,RE6
      REAL LT
      COMMON DUM1(24),IDUM9,G(3),RAD,PI,R,DUM2(9),
     1DUM3(25),H,CR,LT,XC,ACS,ASP,IFLAG
      DATA RELAST/0./,N/0/
C COMPUTE CONSTANTS ONLY ONCE UNLESS RE CHANGES VALUE
      IF (R.EQ.RELAST)  GO TO 2
      RELAST=R
      RE2=R*R
      RE4=RE2*RE2
      RE6=RE4*RE2
      DE=2.0*R
      WRITE(6,341) RELAST,R ,RE2,RE4,RE6,DE
 341  FORMAT(1H ,"RELAST,R ,RE2,RE4,RE6,DE",/,1H ,6E20.6)
      SE=PI*RE2
      SEINV=1.0/SE
      CON1=0.5/(SE*DE)
      PBY3CU=(-RE4)**3
      WRITE(6,342) SE,PI,SEINV,CON1,PBY3CU,XCOF(1)
 342  FORMAT(1H ,"SE,PI,SEINV,CON1,PBY3CU,XCOF(1)",/,1H ,6E20.6)
      XCOF(2)=SE/24.0
      XCOF(3)=0.
      XCOF(4)=-PI*PI/(72.0*SE)
      WRITE(6,343) XCOF(2),XCOF(3),XCOF(4)
 343  FORMAT(1H ,"XCOF(2),XCOF(3),XCOF(4)",/,1H ,6E20.6)
 2     CONTINUE
      XBAR=CG/DE
      PBAR=VZ*CON1
      Q=-576.*PBAR*(PBAR-XBAR)*RE6
      QQ=PBY3CU+(Q*0.5)**2
      WRITE(6,940) Q,QQ,PBY3CU,PBAR,XBAR,RE6
 940  FORMAT(1H ,"Q,QQ,PBY3CU,PBAR,XBAR,RE6",/,1H ,6E20.6)
      IF (QQ.LT.0.)  GO TO 20
      ROOTQ=DSQRT(QQ)
      A=(DABS(-Q*0.5+ ROOTQ  ))**0.33333333
      B=(DABS(-Q*0.5- ROOTQ  ))**0.33333333
      Y1=A+B
      H=DSQRT(Y1)
      IF (IFLAG.EQ.0)  GO TO 30
      Y2=-(A+B)*0.5
      Y3=(A-B)*0.5*1.73205
      WRITE(6,105)  Y1,Y2,Y3
 105  FORMAT(" ",20X,"Y1,Y2,Y3=",3(1PE12.4))
      GO TO 30
 20   CONTINUE
       XCOF(1)=VZ*(VZ*0.5*SEINV-CG)
      N=3
      WRITE(6,9302) XCOF(1)
9302  FORMAT(1X,"XCOF1= ",E12.5)
      CALL POLRT(XCOF,COF,N,ROOTR,ROOTI,IER)
C FIND SMALLEST POSITIVE REAL ROOT
      RMIN=1E10
      DO 10 I=1,N
      IF(ROOTI(I).EQ.0.0.AND.ROOTR(I).LT.RMIN.AND.ROOTR(I).GT.0.) RMIN=
     1   ROOTR(I)
  10  CONTINUE
      IF (IFLAG.EQ.0)  GO TO 6
      DX=1.0/(288.*PBAR)
      XMAX=PBAR+DX
      XMIN=PBAR-DX
C TEST NATURE OF ROOTS
      IF (XBAR.LT.XMAX.AND.XBAR.GT.XMIN.AND.XBAR.GT.PBAR) WRITE(6,101)
      IF (XBAR.LT.XMAX.AND.XBAR.GT.XMIN.AND.XBAR.LT.PBAR) WRITE(6,102)
      IF((XBAR.GT.XMAX.OR .XBAR.LT.XMIN).AND.XBAR.GT.PBAR) WRITE(6,103)
      IF((XBAR.GT.XMAX.OR .XBAR.LT.XMIN).AND.XBAR.LT.PBAR) WRITE(6,104)
 101  FORMAT(" 3 REAL ROOTS, 2 POSITIVE, 1 NEGATIVE",58X,"REAL",6X,
     1 "IMAJ")
 102  FORMAT(" 3 REAL ROOTS, 1 POSITIVE, 2 NEGATIVE",58X,"REAL",6X,
     1 "IMAJ")
 103  FORMAT(" 1 REAL ROOT, NEGATIVE                ",58X,"REAL",6X,
     1 "IMAJ")
 104  FORMAT(" 1 REAL ROOT, POSITIVE                ",58X,"REAL",6X,
     1 "IMAJ")
      WRITE(6,100)XBAR,XMAX,XMIN,PBAR,IER,((ROOTR(I),ROOTI(I)),I=1,N)
 100  FORMAT(1H , "BAR,XMAX,XMIN,PBAR,IER,ROOTS=",4(1PE12.4),I3,8X,
     1 2E12.4,5(/90X,2E12.4))
 6    CONTINUE
C IF NO POSITIVE ROOTS, WRITE ERROR MESSAGE
      IF (RMIN.EQ.1E10)  GO TO 40
      H=SQRT(RMIN)
 30   CONTINUE
      CR=0.5*(H+R *R /H)
      LT=VZ*SEINV-0.5*H-PI*H**3/(6.*SE)
      XC=H+LT-CR
      ASP=PI*CR**2
      ACS=2.*PI*(R *LT+CR*H)
C DO NOT INCLUDE LT IN SURFACE AREA IF IT IS NEGATIVE
      IF (LT.LT.0.)  ACS=2.*PI*CR*H
C CORRECT PROJECTED AREA IF NOT SPHERICAL
      IF (XC.LT.LT)  ASP=SE
      RETURN
 40   CONTINUE
 106  FORMAT("0PROGRAM HALT.  POSITIVE ROOT FOR H NOT FOUND.")
      WRITE(6,100)XBAR,XMAX,XMIN,PBAR,IER,((ROOTR(I),ROOTI(I)),I=1,3)
      WRITE(6,106)
      STOP
      END
      SUBROU:INE RK4
      DIMENSION OLD(11),B(11)
C THIS IS A 4TH ORDER RUNGE-KUTTA INTEGRATOR
      COMMON V(11),DV(11),T,DT,NV,DUCK(46),IDUCK
      DATA J/0/
      OLDT=T
      DO2J=1,NV
    2 OLD(J)=V(J)
      CALL DERIV
      T=OLDT+0.5*DT
      DO4J=1,NV
      B(J)=DT*DV(J)
    4 V(J)=OLD(J)+0.5*B(J)
      CALL DERIV
      DO6J=1,NV
      TMP=DT*DV(J)
      B(J)=B(J)+2*TMP
    6 V(J)=OLD(J)+0.5*TMP
      CALL DERIV
      DO8J=1,NV
      TMP=DT*DV(J)
      B(J)=B(J)+2*TMP
    8 V(J)=OLD(J)+TMP
      T=OLDT+DT
      CALL DERIV
      DO 10 J=1,NV
   10 V(J)=OLD(J)+(B(J)+DT*DV(J))/6
      RETURN
      END
      SUBROUTINE POLRT(XCOF,COF,M,ROOTR,ROOTI,IER)
      DIMENSION XCOF(1),COF(1),ROOTR(1),ROOTI(1)
      DOUBLE PRECISION XO,YO,X,Y,XPR,YPR,UX,UY,V,YT,XT,U,XT2,
     1YT2,SUMSQ,DX,DY,TEMP,ALPHA
      IFIT=0
      N=M
      IER=0
      IF(XCOF(N+1))10,25,10
   10 IF(N) 15,15,32
C   SET ERROR CODE TO 1
   15 IER=1
   20 RETURN
C   SET ERROR CODE TO 4
  25  IER=4
      GO TO 20
```

```
C         SET ERROR CODE TO 2
   30 IER=2
      GO TO 20

   32 IF(N-36) 35,35,30
   35 NX=N
      NXX=N+1
      N2=1
      KJ1=N+1
      DO 40 L=1,KJ1
      MT=KJ1-L+1
   40 COF(MT)=XCOF(L)
C   SET INITIAL VALUES
C
  45  X0=.00500101
      Y0=0.01000101
C          ZERO INITIAL VALUES COUNTER
C
      IN=0
  50  X=X0
C
C          INCREMENT INITIAL VALUES AND COUNTER
C
      X0=-10.0*Y0
      Y0=-10.0*X
C
C          SET X AND Y TO  CURRENT VALUE
C
      X=X0
      Y=Y0
      IN=IN+1
      GO TO 59
  55  IFIT=1
      XPR=X
      YPR=Y
C
C          EVALUATE POLYNOMIAL AND DERIVATIVES
C
  59  ICT=0
  60  UX=0.0
      UY=0.0
      V=0.0
      YT=0.0
      XT=1.0
      U=COF(N+1)
      IF(U) 65,130,65
   65  DO 70 I=1,N
      L=N-I+1
      TEMP=COF(L)
      XT2=X*XT-Y*YT
      YT2=X*YT+Y*XT
      U=U+TEMP*XT2
      V=V+TEMP*YT2
      FI=I
      UX=UX+FI*XT*TEMP
      UY=UY-FI*YT*TEMP
      XT=XT2
 70   YT=YT2
      SUMSQ=UX*UX+UY*UY
      IF(SUMSQ) 75,110,75
 75   DX=(V*UY-U*UX)/SUMSQ
      X=X+DX
      DY=-(U*UY+V*UX)/SUMSQ
      Y=Y+DY
 78   IF(DABS(DY)+DABS(DX)-1.0D-05) 100,80,80
C
C        STEP ITERATION COUNTER
C
 80   ICT=ICT+1
      IF(ICT-500) 60,85,85
 85   IF(IFIT)100,90,100
 90   IF(IN-5) 50,95,95
C
C        SET ERROR CODE TO 3
C
 95   IER=3
      GO TO 20
 100  DO 105 L=1,NXX
      MT=KJ1-L+1
      TEMP=XCOF(MT)
      XCOF(MT)=COF(L)
 105  COF(L)=TEMP
      ITEMP=N
      N=NX
      NX=ITEMP
      IF(IFIT) 120,55,120
 110  IF(IFIT) 115,50,115
 115  X=XPR
      Y=YPR
 120  IFIT=0
 122  IF(DABS(Y)-1.0D-4*DABS(X)) 135,125,125
 125  ALPHA=X+X
      SUMSQ=X*X+Y*Y
      N=N-2
      GO TO 140
 130  X=0.0
      NX=NX-1
 135  Y=0.0
      SUMSQ=0.0
      ALPHA=X
      N=N-1
140   COF(2)=COF(2)+ALPHA*COF(1)
145   DO 150 L=2,N
150   COF(L+1)=COF(L+1)+ALPHA*COF(L)-SUMSQ*COF(L-1)
155   ROOTI(N2)=Y
      ROOTR(N2)=X
      N2=N2+1
      IF(SUMSQ) 160,165,160
160   Y=-Y
      SUMSQ=0.0
      GO TO 155
165   IF(N) 20,20,45
      END
      END$
```

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.
NASA CR-2923

2. GOVERNMENT ACCESSION NO.

3. RECIPIENT'S CATALOG NO.

4. TITLE AND SUBTITLE
Environmental Effects from SRB Exhaust [illegible] – Technique Development and Preliminary Assessm[illegible]

5. REPORT DATE
November 1977

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S) A. I. Goldford, S. I. Adelf[illegible] Hickey, S. R. Smith, R. P. Welty, and G. [illegible] White

8. PERFORMING ORGANIZATION REPORT #
M-236

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Science Applications, Inc.
2109 W. Clinton Avenue, Suite 800
Huntsville, Alabama 35805

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.
NAS8-31806

13. TYPE OF REPORT & PERIOD COVERED
Contractor

12. SPONSORING AGENCY NAME AND ADDRESS
National Aeronautics and Space Administration
Washington, D. C. 20546

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Prepared under the sponsorship of the Atmospheric Diffusion/Environmental Effects Technical Task Team, Dr. J. B. Stephens (Aerospace Environment Division, Space Sciences Laboratory, NASA/Marshall Space Flight Center), Technical Task Leader

16. ABSTRACT

This report documents results of a study which had as its primary objective the development of techniques to determine the environmental effects from the Space Shuttle SRB exhaust effluents. The study developed many new and needed tools which were used to perform a preliminary climatological assessment. This preliminary study will be used to guide the future full-scale climatological assessment. The exhaust effluent chemistry study was performed and, neglecting several possibly important effects, the exhaust effluent species determined. A reasonable exhaust particle size distribution has been constructed for use in future nozzle analyses and for the deposition model. The effects of scavenging and absorption were not included in the preliminary assessment. The preliminary assessment was used to identify problems that may be associated with the full-scale assessment; therefore, these preliminary air quality results should be used with caution in drawing conclusions regarding the environmental effects of the Space Shuttle exhaust effluents.

17. KEY WORDS
Rocket Motor Exhaust Diffusion
Rocket Exhaust Cloud Rise
Space Shuttle Environmental Studies
Atmospheric Diffusion
REED Description

18. DISTRIBUTION STATEMENT
Category 34

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	340	$10.00

For sale by the National Technical Information Service, Springfield, Virginia 22161

*U.S. GOVERNMENT PRINTING OFFICE: 1977 - 735-078/49

END

5.12.78